

PARAHUARI SOLNOWSKI BRANCO

IMPLEMENTAÇÃO DE UMA REDE SEM FIO: ESTUDO DE CASO NA PLATAFORMA WINDOWS

Dissertação apresentada como requisito parcial
à obtenção do grau de Mestre em Informática
no P. P. G. Informática da Universidade Federal
do Paraná.

Orientador: Carlos A. Picango de Carvalho

CURITIBA

2000

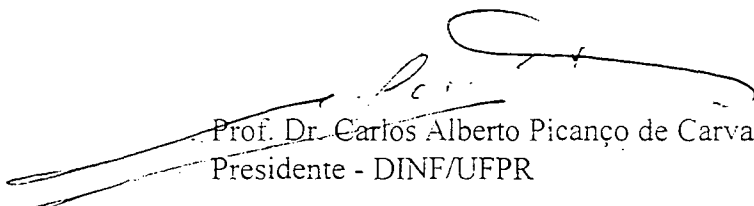


Ministério da Educação
Universidade Federal do Paraná
Mestrado em Informática


PARECER

Nós, abaixo assinados, membros da Comissão Examinadora da defesa de Dissertação de Mestrado em Informática, do aluno Parahuari Solnowski Branco, avaliamos o trabalho intitulado **“Implementação de uma Rede sem Fio: Estudo de um Caso no Sistema Operacional Windows”**, cuja defesa foi realizada no dia 10 de maio de 2000. Após a Avaliação, decidimos pela Aprovação do Candidato.


Curitiba, 10 de maio de 2000.



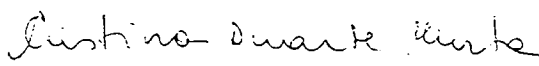
Prof. Dr. Carlos Alberto Picanço de Carvalho
Presidente - DINF/UFPR



Prof. Dr. Wilson Arnaldo Artuzi Jr.
Membro Externo - ELETR/UFPR



Prof. Dra. Keiko Verônica Ono Fonseca
Membro Externo - CEFET-PR/ CPGEI



Prof. Dra. Cristina Duarte Murta
Membro - DINF/UFPR

PARAHUARI SOLNOWSKI BRANCO

**IMPLEMENTAÇÃO DE UMA REDE SEM FIO: ESTUDO DE CASO NO
SISTEMA OPERACIONAL WINDOWS**

CURITIBA

2000

AGRADECIMENTOS

Ao meu orientador Carlos Picanço, João Fábio, Paulo Otsuka e ao Rogério da Logitech pelo apoio na realização deste trabalho.

A professora Cristina Duarte pela paciência e compreensão.

A professora Keiko Fonseca por suas sugestões.

Aos professores Elias e Sílvia pelas sugestões apresentadas no início deste trabalho.

A toda a minha família pelo carinho, apoio e compreensão.

SUMÁRIO

LISTA DE FIGURAS	X
RESUMO	XII
ABSTRACT	XIII
1. INTRODUÇÃO.....	1
1.1 MOTIVAÇÃO: QUAL É O PROBLEMA?.....	1
1.2 A DISSERTAÇÃO	2
2. COMUNICAÇÃO SEM FIO - HISTÓRIAS E TECNOLOGIAS	4
2.1 INVENÇÃO DO RÁDIO	4
2.2 LINHA DO TEMPO	6
2.3 REDES SEM FIO	8
2.3.1 O que é?	8
2.3.2 Por que utilizar uma rede sem fio?	9
2.3.3 Como as redes sem fio são utilizadas no mundo real?	11
2.3.4 Preocupações sobre o uso da comunicação sem fio.....	13
2.4 TECNOLOGIAS PARA COMUNICAÇÃO SEM FIO	15
2.4.1 Comunicação Celular	15
2.4.2 Celular Analógico.....	16
2.4.3 CDPD (Cellular Digital Packet Data)	17
2.4.4 TDMA (Time Division Multiple Access).....	17
2.4.5 GSM (Global System for Mobile Communication)	18
2.4.6 Outros tipos de comunicação.....	18
2.5 COMO UMA REDE SEM FIO FUNCIONA	18
2.6 MERCADO ATUAL	20
2.6.1 Comunicação de dados via celular	20

2.6.2 Acesso à Web via celulares, pagers e computadores de mão	23
2.6.3 Redes locais sem fio	25
2.7 FUTURO DO MERCADO SEM FIO	26
3. PESQUISAS NA ÁREA.....	29
3.1 INTRODUÇÃO	29
3.2 O SISTEMA ALOHA DA UNIVERSIDADE DO HAVAI	29
3.3 ÁREAS DE PESQUISA NA COMPUTAÇÃO SEM FIO	31
3.3.1 Desconexão.....	33
3.3.2 Largura de Banda	34
3.3.3 Grande Variabilidade na Largura da Banda.....	36
3.3.4 Redes Heterogêneas.....	36
3.3.5 Riscos de Segurança	37
3.3.6 Migração de Endereço	38
3.3.7 Difusão Seletiva	38
3.3.8 Serviços Centrais.....	39
3.3.9 Home Bases	39
3.3.10 Transferência de Ponteiros.....	39
3.3.11 Informações Dependentes de Localização.....	40
3.3.12 Low Power	40
3.3.13 Risco para os Dados	40
3.3.14 A Confiabilidade na Transmissão	41
3.4 ESPECIFICAÇÃO BLUETOOTH	41
4. MODELO PROPOSTO.....	43
4.1 INTRODUÇÃO	43
4.2 MODELO DE REFERÊNCIA.....	43
4.2.1 O modelo OSI	45
4.2.2 O Padrão IEEE 802	46

4.2.3 O padrão IEEE 802.11	48
4.2.4 IEEE 802.11 Wireless Local Area Network.....	48
4.2.5 Topologia	50
4.2.5.1 IBSS.....	50
4.2.5.2 ESS	51
4.3 ARQUITETURA DE REDE DO WINDOWS	51
4.3.1 NDIS Drivers.....	54
4.3.1.1 NIC Drivers.....	55
4.3.1.2 Intermediate protocol drivers.....	55
4.3.1.3 Upper level protocol driver.....	56
4.4 COMPONENTES "DE PRATELEIRA"	56
4.5 RESTRIÇÕES.....	57
4.6 ARQUITETURA DA SOLUÇÃO	58
4.7 DESENVOLVENDO O NDIS MINIPORT NIC DRIVER	61
4.7.1 Nível Físico - NIC	61
4.7.2 Rotinas para a manipulação do NIC.....	62
4.7.3 Delimitação dos Pacotes	62
4.7.4 Nível de Enlace – Controle de Acesso.....	63
4.7.5 Controle de Erros/Fluxo.....	65
4.7.5.1 CRC	66
4.7.5.2 Pacotes especiais.....	66
4.7.5.3 Pacotes de dados.....	68
4.7.6 Topologia	69
5. IMPLEMENTAÇÃO.....	70
5.1 COMO A PLACA FUNCIONA?	72
5.3 COMUNICANDO ATRAVÉS DE CABOS.....	73
5.4 ENTENDENDO O FUNCIONAMENTO DE UM NDIS MINIPORT NIC DRIVER.....	75
5.4.1 NDIS Miniport NIC driver.....	75
5.4.2 Código fonte NE2000.....	77

5.5 ENVIANDO E RECEBENDO PACOTES	78
5.6 PRIMEIRA VERSÃO DO NDIS MINIPOORT NIC DRIVER	79
5.7 ENVIANDO E RECEBENDO PACOTES ATRAVÉS DO RÁDIO	81
5.8 SEGUNDA VERSÃO DO NDIS MINIPOORT NIC DRIVER	82
5.8.1 <i>Teste através do ambiente de rede</i>	84
5.8.2 <i>Comunicação utilizando o padrão Internet</i>	85
5.9 TERCEIRA VERSÃO DO NDIS MINIPOORT NIC DRIVER	86
6. CONCLUSÃO E TRABALHOS FUTUROS	89
APÊNDICE A - DESCRIÇÃO DO HARDWARE UTILIZADO NO PROJETO.....	91
A1 PLACA BAY 96 USCC	92
A2 Z8530.....	93
A2.1 <i>Registradores</i>	93
A2.2 <i>Modos de Operação</i>	94
A2.2.1 <i>Modo Assíncrono</i>	94
A2.2.2 <i>Modo Síncrono</i>	95
A2.2.3 <i>Modo SDLC</i>	95
A3 PINAGEM	95
A4 USO COM O RÁDIO.....	96
A5 ENDEREÇAMENTO	96
A6 ACESSO AOS REGISTRADORES.....	99
A7 PROGRAMAÇÃO DA USCC	102
A7.1 <i>Polling</i>	102
A7.2 <i>Interrupção</i>	102
A8 INICIALIZAÇÃO DA PLACA	103
A9 CONTROLE DAS INTERRUPÇÕES	107
A9.1 <i>Recepção</i>	108
A9.2 <i>Transmissão</i>	108

A9.3 <i>External/Status</i>	109
A9.4 <i>Identificando o motivo da interrupção</i>	109
A10 ENVIO DE DADOS	111
A10.1 <i>Polling</i>	112
A10.2 <i>Interrupção</i>	113
A11 RECEPÇÃO DE DADOS	113
A11.1 <i>Polling</i>	114
A11.2 <i>Interrupção</i>	114
A12 MODO LOOPBACK	115
APÊNDICE B - ETHERNET	116
B1 ETHERNET	116
B2 CAMADAS ETHERNET	117
B3 ETHERNET X IEEE 802.3	117
B4 FORMATO DO PACOTE	119
B5 ENDEREÇO ETHERNET	120
B6 CÓDIGOS NT	121
APÊNDICE C - FERRAMENTAS UTILIZADAS NO PROJETO	122
C1 MICROSOFT SDK/DDK	122
C2 SOFTICE NUMEGA	122
C3 BORLAND C++ 3.1	123
C4 VISUAL C++ 6.0	124
C5 WORD 97	124
C6 NDISCOPE FOR WINDOWS	124
C7 BAYCOM	124
C8 JNOS	125
C9 SOURCER	125
C10 ADOBE ACROBAT READER	125

C11 GSVIEW32 2.1 - A GHOSTSCRIPT GRAPHICAL INTERFACE.....	125
C12 TERMINAL DO WINDOWS	126
APÊNDICE D - SCRIPT PARA INSTALAÇÃO DO DRIVER DE REDE	127
APÊNDICE E - REPRODUÇÃO DO EXPERIMENTO PRÁTICO	129
REFERÊNCIAS BIBLIOGRÁFICAS	130

LISTA DE FIGURAS

FIGURA 2.1 — EXEMPLO DE UM AMBIENTE DE REDE SEM FIO USANDO PONTOS DE ACESSO.....	20
FIGURA 3.1 — ELEMENTOS BÁSICOS DO SISTEMA ALOHA.....	30
FIGURA 3.2 — COLISÃO NA TRANSMISSÃO DO SISTEMA ALOHA.....	30
FIGURA 3.3 — DIAGRAMA DE TRANSMISSÃO E RECEPÇÃO DO SISTEMA ALOHA.....	31
FIGURA 4.1 — USANDO O SISTEMA OPERACIONAL PARA DISPONIBILIZAR ACESSO TRANSPARENTE A REDE SEM FIO	45
FIGURA 4.2 — MODELO OSI X MODELO IEEE.....	47
FIGURA 4.3 — PADRÃO 802.11 ACRESCENTADO AO IEEE.....	48
FIGURA 4.4 — MEIOS FÍSICOS ABRANGIDOS PELO PADRÃO 802.11.....	48
FIGURA 4.5 — IBSS O TIPO MAIS BÁSICO DE REDE SEM FIO 802.11	51
FIGURA 4.6 — ESS SATISFAZ A NECESSIDADE DAS REDES DE GRANDE ABRANGÊNCIA E MAIOR COMPLEXIDADE	51
FIGURA 4.7 — MODELO OSI E A ARQUITETURA DE REDE DO WINDOWS.....	52
FIGURA 4.8 — COMPONENTES EXISTENTES NA ARQUITETURA DE REDE DO WINDOWS	54
FIGURA 4.9 — ARQUITETURA GERAL DA SOLUÇÃO	60
FIGURA 4.10 — COMPONENTES A SEREM PROGRAMADOS NO NDIS MINIPORT NIC DRIVER	61
FIGURA 4.11 — IDENTIFICAÇÃO DO PACOTE	62
FIGURA 4.12 — RESUMO DO FUNCIONAMENTO DO CSMA/CA.....	64
FIGURA 4.13 — DIAGRAMA DAS OPERAÇÕES DE TRANSMISSÃO E RECEPÇÃO.....	67
FIGURA 5.1 — PROGRAMA BAYCOM CONECTANDO COM OUTRO COMPUTADOR (PU1JIF) UTILIZANDO CABOS	74
FIGURA 5.2 — PROGRAMA DOS QUE TROCA BYTES ENTRE DOIS COMPUTADORES CONECTADOS COM A PLACA LIGADA ATRAVÉS DE UM CABO	75
FIGURA 5.3 — PACOTE ETHERNET	78
FIGURA 5.4 — PACOTE USADO NO EXPERIMENTO.....	79
FIGURA 5.5 — ORGANIZAÇÃO DA PRIMEIRA VERSÃO DO NDIS MINIPORT	80
FIGURA 5.6 — PROGRAMA DOS QUE TROCA PACOTES ENTRE DOIS COMPUTADORES LIGADOS POR RÁDIO	82
FIGURA 5.7 — ORGANIZAÇÃO DA SEGUNDA VERSÃO DO NDIS MINIPORT	83

FIGURA 5.8 — SEQUÊNCIA DE PACOTES PARA A COMUNICAÇÃO DE DADOS	83
FIGURA 5.9 — EXCLUÍDO OS PACOTES DE CONTROLE OPCIONAIS NESTA VERSÃO DO DRIVER.....	87
FIGURA 5.10 — TESTE DO FUNCIONAMENTO DO DRIVER COM APLICAÇÕES INTERNET	87
FIGURA 5.11 — APLET JAVA COM UM PROGRAMA DE CHAT SIMPLES.....	88
FIGURA A.1 — PINAGEM LIGANDO A PLACA USCC AO RÁDIO.....	96
FIGURA A.2 — <i>JUMPER</i> PARA A SELEÇÃO DO ENDEREÇO BASE DA PLACA	97
FIGURA A.3 — ENDEREÇO DE DADOS E DE CONTROLE DA PLACA USCC USANDO COMO ENDEREÇO BASE O VALOR 300H	97
FIGURA A.4 — PROGRAMAÇÃO DOS REGISTRADORES PARA A INICIALIZAÇÃO DA PLACA	105
FIGURA B.1 — CAMPOS DO PACOTE ETHERNET	120
FIGURA D.1 - JANELA DO WINDOWS 98 PARA ADICIONAR O DRIVER DA REDE SEM FIO	127

RESUMO

Esta dissertação é o resultado do estudo da viabilização técnica e prática da comunicação sem fio entre microcomputadores mediante a utilização de tecnologia e instrumentos "de prateleira". Como resultado deste estudo é apresentado um levantamento da área de comunicação de dados sem fio contendo suas características, as tecnologias utilizadas e suas áreas de pesquisa.

O texto é finalizado com a proposta de uma arquitetura de hardware e software para a implementação de uma rede sem fio. Esta proposta teve como base o estudo do modelo OSI, do padrão IEEE 802.11 e da arquitetura de rede da plataforma Windows. A solução apresentada está baseada no desenvolvimento de um *NDIS Miniport NIC driver* e na utilização de ferramentas Internet para o desenvolvimento das aplicações que utilizam esta rede sem fio.

ABSTRACT

This dissertation is the result of a technical viability study of wireless communication through computers using "on-the-shelf" technology and instruments. As a result of this study, information about the wireless communication field, its characteristics, technologies and research areas is presented.

This text also presents an architecture of hardware and software to develop a wireless network. This proposal is based in the OSI model, IEEE 802.11 specification and Windows network architecture. The solution is based in the development of a NDIS Miniport NIC driver and in the use of Internet tools to the application development using this wireless network.

1. INTRODUÇÃO

1.1 MOTIVAÇÃO: QUAL É O PROBLEMA?

Existe um segmento de mercado onde a informática tem sido usada para a monitoração de equipamentos de rádio e televisão. Nesta área a concorrência, principalmente de sistemas americanos, é muito grande. Por isso, para conseguir competir neste mercado é necessário atacar os pontos fracos da maioria destes sistemas de monitoração: a sua arquitetura fechada, o seu alto custo de aquisição, manutenção e personalização. Focando estes pontos, foi desenvolvido um trabalho em parceria com a Rede Globo de Televisão - São Paulo, com o objetivo de construir estes sistemas utilizando soluções mais "abertas" de hardware e de software. No lugar de usar hardware do tipo "caixa-preta", presente nos sistemas importados, foram utilizados computadores padrão PC. O software para fazer a monitoração foi desenvolvido com opções simples de personalização de forma a permitir o usuário do sistema modificar, retirar ou acrescentar facilmente pontos de monitoração. Dentro desta estratégia foi possível reduzir os custos de implantação e personalização destes sistemas.

O objetivo de baixar o custo dos sistemas de supervisão para equipamentos de rádio e televisão foi alcançado, mas o componente responsável pela comunicação dos dados via rádio, que consumiu um considerável tempo de desenvolvimento, apresentou o mesmo problema dos sistemas concorrentes. A comunicação por rádio foi desenvolvida dentro de uma arquitetura fechada, de forma a atender apenas a comunicação existente no sistema de monitoração remota. Um fator não foi considerado no desenvolvimento do sistema de

monitoração: a possibilidade de reaproveitar a solução de comunicação de dados via rádio em outros sistemas que se beneficiem deste tipo de comunicação.

Baseado na experiência adquirida no desenvolvimento destes programas a seguinte pergunta surgiu:

Como desenvolver uma solução, de baixo custo, que permitisse a comunicação de dados via rádio e que fosse transparente ao desenvolvedor do sistema?

Um módulo que permitisse a sua utilização em um sistema de monitoração bem como em outros sistemas que utilizam comunicação de dados sem fio? O enfoque do primeiro trabalho (sistema de telesupervisão) foi a simplificação tendo como alvo o usuário final. O enfoque desta dissertação de Mestrado é diferente, consiste na simplificação do uso da comunicação de dados sem fio para os desenvolvedores de sistemas. Isto é formalizado no parágrafo a seguir.

Este trabalho é o resultado do estudo da viabilização técnica e prática da comunicação sem fio entre microcomputadores mediante a utilização de tecnologia e instrumentos "de prateleira". Entende-se "de prateleira" equipamentos simples, de baixo custo, encontrados em lojas especializadas.

1.2 A Dissertação

O resultado da pesquisa para o desenvolvimento de um módulo de comunicação sem fio está registrado nesta dissertação, organizada da seguinte forma:

- **Primeiro.** Levantamento da área de comunicação de dados sem fio, suas características e tecnologias (capítulo 2). Deve ser lido por quem deseja ter uma visão geral da área.
- **Segundo.** Identificação das áreas de estudo sobre o assunto (capítulo 3). Deve ser lido por quem tem interesse em conhecer o grande número de problemas que existem na área de comunicação sem fio.
- **Terceiro.** Após a apresentação geral sobre a comunicação sem fio (capítulo 1 e 2) está sendo proposto uma arquitetura de hardware e software capaz de atender as necessidades de comunicação da rede sem fio de baixo custo (capítulo 4). Deve ser lido por quem tem interesse no modelo para a implementação desta rede, objeto de estudo da dissertação.
- **Quarto.** Validação do modelo através da implementação de um protótipo (capítulo 5 e apêndices).

De forma a contribuir com trabalhos futuros, além da apresentação do modelo proposto para a rede, foi reunido neste texto as informações sobre a comunicação sem fio encontradas de maneira dispersa ou que não estavam devidamente documentadas. Um exemplo disso é o Apêndice A, que apresenta instruções sobre como programar uma placa USCC.

2. COMUNICAÇÃO SEM FIO - HISTÓRIAS E TECNOLOGIAS

2.1 Invenção do rádio

A invenção do telégrafo por Samuel F. B. Morse em 1838 inaugurou uma nova época nas comunicações. Nos primeiros telégrafos utilizados no século XIX, mensagens eram codificadas em cadeias de símbolos binários, chamados de código Morse, e então transmitidas manualmente por um operador através de um dispositivo gerador de pulsos elétricos. Desde então, a comunicação através de sinais elétricos atravessou uma grande evolução, dando origem à maior parte dos grandes sistemas de comunicação que existem hoje em dia, como o telefone, o rádio e a televisão [2].

Durante a metade do último século os cientistas estavam procurando uma forma de se comunicar sem a utilização de um fio. Todavia, as tentativas feitas com ondas eletromagnéticas não tiveram nenhum resultado. O escocês Clark Maxwell demonstrou matematicamente como “ações” eletromagnéticas se espalhavam com um movimento ondulatório. Em 1887 o alemão Heinrich Hertz, usando correntes periódicas em uma alta frequência, demonstrou a existência real de ondas eletromagnéticas transformando o “movimento ondulatório” em um fenômeno estacionário, o qual poderia ser verificado em um laboratório. O italiano Augusto Righi continuou e melhorou o trabalho de Hertz mostrando a relação entre vibrações óticas e elétricas. Em 1884 Temistocle Calzecchi-Onesti observou a influência de descargas elétricas de perturbações atmosféricas em um *iron filing* construindo um “tubo” que foi chamado mais tarde em 1894 por Oliver Lodge de *coherer*. Foram estas pesquisas que levaram Marconi a produzir um dispositivo capaz de produzir ondas eletromagnéticas e permitir que estas ondas viajassem pelo ar. [41]

Em Setembro de 1895 Guglielmo Marconi, um auto-didata de 21 anos nascido em Bologna, Itália, já tinha feito alguns experimentos simples que o convenceram de que era possível enviar sinais através do uso de ondas eletromagnéticas usando uma antena. Usando os osciladores de Hertz e Righi, Marconi conseguiu fazer pequenas transmissões, como por exemplo uma transmissão que percorreu os 100 metros que separavam a sua casa do final do seu jardim. Todavia, foi possível demonstrar que usando o ar (*ether*) era possível transmitir entre dois pontos separados por um obstáculo. Cientistas e outros especialistas diziam que ondas eletromagnéticas poderiam apenas ser transmitidas em linha reta e que isso funcionaria apenas se não existisse nenhum obstáculo no caminho. Acima de tudo, os cientistas achavam que o obstáculo intransponível à transmissão era a própria curvatura da superfície terrestre. Mesmo sabendo de tudo isso Marconi resolveu colocar o seu transmissor perto de sua casa e o receptor a 3 Km de distância, atrás de um morro. No receptor estava Mignani, um assistente de Marconi, cuja tarefa seria disparar um rifle caso recebesse algum sinal. Quando Mignani disparou a sua arma, pela primeira vez na história os três pontos que compõem a letra “S” do código Morse tinha viajado através do espaço. Mesmo com essa façanha notável os italianos não se entusiasmaram com a invenção de Marconi, o ministro responsável pelas comunicações na Itália considerou que este invento não era “apropriado para as telecomunicações”. Isso levou Marconi à Inglaterra onde patenteou o seu invento. Foi ainda na Inglaterra (1897) que Marconi conseguiu apoio e dinheiro para continuar os seus experimentos, onde conseguiu transmitir a distâncias de 5, 8, 15, 30 e 100 Km.[40]

O ano de 1921 marcou o primeiro registro de uso de comunicações sem fio na forma bidirecional pelo Departamento de Polícia de Detroit. Mais de 20 anos depois, em 1946,

tocou o primeiro telefone móvel [27]. Até recentemente a conectividade sem fio e onipresente parecia filme de ficção científica.

Para resumir a história da comunicação sem fio são apresentados alguns aspectos interessantes, alguns não mencionado em detalhes, através de uma linha do tempo.

2.2 Linha do Tempo

- 1838 Invenção do telégrafo por Samuel F. B. Morse.
- 1850 Clark Maxwell demonstra matematicamente como ações eletromagnéticas espalham-se como um movimento ondulatório.
- 1887 Heinrich Hertz demonstra a real existência de ondas eletromagnéticas.
- 1894 *The Coherer*.
- 1895 Rádio-transmissor de Marconi.
- 1921 Comunicação móvel sem fio bidirecional pelo Departamento de Polícia de Detroit.
- 1941 Patenteado por Hedy Lamarr e George Antheil sistema de comunicação utilizando múltiplas frequências (futuramente chamado de *spread-spectrum*).
- 1946 Funcionamento do primeiro telefone móvel.
- 1962 Primeiro sistema eletrônico de *spread-spectrum* desenvolvido para equipar míssil americano (Sylvania Electronic Systems).

- 1970** Rede de radiodifusão Aloha inicia o seu funcionamento interligando o centro de computação da Universidade do Havaí, em Honolulu, a terminais espalhados pelas ilhas.
- 1980** Em Fevereiro foi instituído o comitê IEEE 802 (chamado assim devido ao ano e mês do seu início), com o objetivo de elaborar padrões para redes locais de computadores.
- 1980s** Rádio amadores desenvolvem um hardware chamado TNC (*Terminal Node Controllers*). Este hardware permitiu que microcomputadores dos Estados Unidos e do Canadá permanecessem conectados através de equipamentos de rádio amador.
- 1984** A ISO desenvolve um modelo de referência chamado OSI (*Open System Interconnection*).
- 1985** O FCC (*Federal Communications Commission*) autoriza o uso público das frequências que compõem o ISM (*Industrial, Scientific and Medical*), esta faixa de frequências compreendem de 902MHz a 5.85 GHz. A autorização permite a utilização de equipamentos que transmitem nesta faixa sem o pedido de autorização do governo.
- 1991** Em Maio um grupo liderado por Victor Hayes envia um projeto ao IEEE para dar início ao grupo de trabalho 802.11 que define o padrão para redes sem fio.
- 1992** Na conferência de administração de espectro de rádio da UIT, instala-se a IMT-2000 (*International Mobile Telecommunications 2000*). A proposta da

IMT-2000 é fornecer acesso *wireless* à infra-estrutura global de telecomunicações, mediante unificação dos padrões adotados em sistemas móveis celulares em uma única infra-estrutura de rádio.

1994 De maneira a incentivar o desenvolvimento da superinfovia da comunicação, que também inclui a infraestrutura *wireless*, o Grupo dos 7 (G-7) cria a GII (Global Information Infrastructure).

1997 Em Junho o texto descrevendo o padrão IEEE 802.11 chega a sua versão final.

1999 Em Março foi criado o grupo de trabalho IEEE 802.15 para definir os padrões para a comunicação de dispositivos em redes do tipo WPAN (*Wireless Personal Area Network*).

2.3 Redes sem fio

2.3.1 O que é?

Uma rede local sem fio é um sistema de comunicação de dados flexível implementado como uma extensão, ou uma alternativa, a uma rede local (LAN) [51]. Usando tecnologia de rádio frequência (RF), redes sem fio transmitem e recebem dados através do ar, minimizando a necessidade de conexões usando cabo. Desta forma, as redes sem fio combinam conectividade com mobilidade.

As redes sem fio têm aumentado a sua popularidade em diversas áreas: saúde, manufatura, vendas e educação. Estas áreas têm tido resultados positivos no uso de terminais de entrada de dados e computadores portáteis na transmissão de informações em

tempo real para a um computador central. Hoje as redes sem fio estão sendo reconhecidas como uma alternativa de conexão para redes de propósito geral em uma grande faixa de negócios. Pesquisas estão prevendo que este mercado aumentará o seu ritmo de crescimento, devendo ultrapassar a barreira dos US\$2 bilhões a partir do ano 2000 [28] [38].

2.3.2 Por que utilizar uma rede sem fio?

As redes locais sem fio são realidade em vários ambientes de redes, principalmente nos que requerem a mobilidade dos usuários. As aplicações são as mais diversas e abrangem desde aplicações médicas, como por exemplo a visita a vários pacientes com um sistema portátil de monitoramento, até ambientes de escritório ou de fábrica.

Apesar das limitações de cobertura geográfica, utilizando-se a arquitetura de sistemas de distribuição pode-se aumentar a abrangência de uma rede sem fio, fazendo uso de vários sistemas de distribuição interconectados via rede com fio, num esquema de *roaming* entre microcélulas, semelhante a um sistema de telefonia celular convencional.

A disseminação do uso de rede nos negócios, o crescimento meteórico da Internet e dos serviços *online* são os mais fortes indícios das vantagens do compartilhamento de dados e recursos. Através das redes sem fio, os usuários podem acessar informações compartilhadas sem ter que procurar por um lugar para se conectar. Os gerentes de rede podem configurar ou aumentar as redes sem a necessidade de instalação ou movimentação de cabos. Redes sem fio oferecem produtividade, conveniência e vantagens de custo sobre as redes que utilizam cabos. Listamos algumas vantagens do uso das redes sem fio:

- **Mobilidade:** sistemas de rede sem fio podem fornecer aos usuários da rede acesso a informações em tempo real em qualquer lugar dentro da organização. Esta possibilidade de movimentação pode ser utilizada de forma a aumentar a produtividade, além de criar novas oportunidades de serviços que não seriam possíveis através do uso de redes com fio.
- **Velocidade de Instalação e Simplicidade:** instalar uma rede sem fio pode ser rápido e fácil, eliminando a necessidade de passar cabos pelos tetos e paredes.
- **Flexibilidade na Instalação:** redes sem fio podem ir onde o sistema de cabos não pode, como por exemplo em uma reserva florestal.
- **Custo de Propriedade:** enquanto o custo inicial de investimento necessário para a instalação do hardware e do software pode ser maior do que o custo de uma rede com cabos, o custo total de instalação e o custo do ciclo de vida pode ser significativamente inferior. Os benefícios a longo prazo são maiores para ambientes dinâmicos que necessitam de mudanças frequentes. As redes com cabos precisam de modificações em sua estrutura, como instalação de novos cabos e novos pontos na rede.
- **Escalabilidade:** sistemas de rede sem fio podem ser configurados numa grande variedade de topologias para atender as necessidades de aplicações e instalações específicas. Configurações são facilmente modificadas de redes ponto a ponto de poucos usuários para redes corporativas, permitindo uma movimentação do usuário em uma grande área de abrangência.

2.3.3 Como as redes sem fio são utilizadas no mundo real?

A adoção de uma arquitetura de rede que utiliza comunicação sem fio não significa necessariamente a exclusão da utilização de cabos na rede. Muitas vezes uma rede sem fio é utilizada de forma a complementar ao invés de substituir uma rede tradicional utilizando cabos. Frequentemente a rede sem fio é utilizada de forma a permitir a conexão de um usuário móvel à rede corporativa que utiliza cabos. A lista apresentada a seguir descreve algumas das muitas aplicações possíveis que tiram proveito da flexibilidade oferecida pela comunicação sem fio:

- Médicos e enfermeiras em um hospital são muito mais produtivos porque através de um *handheld* ou um *notebook* podem obter informações atualizadas sobre o paciente instantaneamente.
- Gerentes de rede gastam menos tempo para viabilizar mudanças na rede, como posição dos computadores, local ou tamanho da rede.
- Locais de treinamento e estudantes em universidades utilizam conexões sem fio para simplificar o acesso à informação e à troca de dados, facilitando o aprendizado.
- Gerentes de rede que instalam computadores em construções antigas descobriram que a utilização de redes sem fio é uma alternativa de infraestrutura de rede que pode ficar mais barata.
- A utilização de redes sem fio em congressos, feiras ou eventos reduzem o tempo de instalação porque diminui a inconveniente instalação dos cabos e porque permite que os computadores sejam pré-configurados e testados, mesmo antes da instalação em seu lugar definitivo.

- Gerentes de rede que utilizam redes sem fio podem aumentar a segurança de aplicações de missão crítica a partir da criação de formas alternativas de conexão ao ambiente de rede com cabos.
- Executivos em reuniões podem tomar decisões mais rapidamente porque possuem informações em tempo real na "ponta dos dedos". [51]
- Interligação de LANs entre edifícios independentes ou distantes.
- Instalações temporárias.
- Prédios tombados pelo patrimônio histórico.
- Locais onde já existem diversos dutos instalados no solo, muitas vezes sem um controle da sua exata localização (Sabesp, Eletropaulo, Telepar, entre outras).
- Usuários que precisam se locomover dentro da empresa, e ao mesmo tempo consultam ou atualizam informações existentes na rede da empresa, como por exemplo o inventário, o estoque, entre outros. [28]
- Correio eletrônico para equipamentos portáteis.
- Telemetria de equipamentos em locais remotos ou de difícil acesso, como retransmissores de TV.
- Acompanhamento de frotas através de localização e telemetria.
- Acesso a base de dados remota (polícia, guarda florestal, sistemas do tipo *rede-shop*, entre outras).

- Máquinas vendedoras automáticas com modems sem fio que avisam quando o estoque está acabando, dispensando o deslocamento de pessoas para a verificação.

[26]

2.3.4 Preocupações sobre o uso da comunicação sem fio

São muitos os benefícios que podem ser alcançados pelo uso da comunicação sem fio, mas não se pode esquecer dos problemas e consequência da escolha desta tecnologia.

- **Interferência no sinal de rádio.** O processo de transmissão e recepção de sinais de rádio ou laser através do ar faz com que os sistemas sem fio sejam vulneráveis a ruídos produzidos por mudanças atmosféricas ou transmissões realizadas por outros sistemas. Uma rede sem fio pode também interferir em uma outra rede sem fio próxima ou em um outro equipamento de rádio. Ou seja, uma comunicação sem fio pode sofrer ou gerar interferências. O gerenciamento das frequências utilizadas, uma boa análise do espectro antes da instalação da rede e a utilização de tecnologias como *spread spectrum* podem minimizar este tipo de problema.

- **Gerenciamento de Energia.** Na utilização de equipamentos móveis, muitas vezes não é possível ligar este aparelho a uma tomada. Neste caso a autonomia de uso do equipamento estará dependendo de sua bateria. Nestas situações o gasto adicional resultante da utilização do hardware responsável pela comunicação sem fio, passa a ser significativo no consumo de energia. Uma possível alternativa para reduzir o consumo de bateria está na utilização de técnicas que, mesmo aumentando o tempo de resposta, procuram reduzir a quantidade de tempo que o rádio permanece em operação.

- **Interoperabilidade dos sistemas.** Diferentemente das redes com fio que possuem protocolos altamente difundidos, como por exemplo Ethernet, a estruturação de uma rede sem fio com um certo tipo de fabricante pode inviabilizar a ampliação da rede utilizando equipamentos de outros fabricantes. A publicação e a atual adoção de padrões pelos fabricantes (como o IEEE 802.3) deverá aumentar a compatibilidade destes equipamentos em poucos anos.

- **Segurança da rede contra acesso não autorizados.** O ar é usado como meio de transporte para as informações, por isso a possibilidade de ter acesso aos dados transmitidos é maior se comparado à comunicação utilizando cabos. Por isso, além do controle das conexões e do uso de senhas, a utilização de criptografia é *fundamental*.

- **Questões de instalação.** Em comparação às redes sem fio o planejamento para a instalação dos cabos de uma rede com fio normalmente não apresenta problemas. Estuda-se a planta, identificam-se os locais para passagem dos cabos, o comprimento dos cabos e a eventual necessidade de repetidores. Depois da instalação destes cabos a rede costuma funcionar de acordo com o planejado. De forma contrária, a instalação de redes sem fio não costuma ser tão previsível. É muito difícil, se não impossível, definir a estruturação da rede sem fio apenas estudando a planta ou observando o local da instalação. Paredes, tetos e objetos atenuam os sinais de rádio e podem até alterar o caminho da transmissão. A abertura ou fechamento de uma porta pode alterar a propagação do sinal, sem contar a possibilidade de interferência externa quando o sinal é utilizado na comunicação entre prédios diferentes, que neste caso pode receber interferência de outras

transmissões. Por isso é fundamental a realização de testes de propagação e análise do espectro como parte do planejamento de uma rede sem fio.

2.4 Tecnologias para Comunicação sem Fio

Pode-se classificar a comunicação sem fio em comunicações unidirecionais e bidirecionais. As comunicações unidirecionais, presentes nos serviços de *paggers*, permitem transmissão de informações para um ou para múltiplos receptores. Este é o formato ideal para o despacho de mensagens, atualização de notícias, informações de preços e atribuição de rotas. Considerando custo, a comunicação unidirecional é um método eficiente para a entrega de informações aos seus destinatários.

Embora as transmissões unidirecionais desempenhem um importante papel são as comunicações bidirecionais que permitem explorar o potencial máximo da comunicação sem fio, possibilitando conexão de LANs, telefones públicos sem fio, conferências em tempo real, entre outras. [27]

Existe uma série de tecnologias para comunicação sem fio, cada uma destas tecnologias com o seu conjunto de características, vantagens e limitações. Apresentamos estas tecnologias a seguir.

2.4.1 Comunicação Celular

Fazem parte de um sistema móvel de radiotelefonia que transmite dados sobre pequenas áreas geográficas, chamadas células, que utilizam receptores e transmissores de baixa potência. Os telefones celulares funcionam tipicamente dentro de um raio de 8km. A comunicação celular ocorre por meio de uma malha de transmissores/receptores, onde cada

conjunto é chamado de célula. Um sinal proveniente de um dispositivo celular é tratado pela célula mais próxima, que transmite o sinal através de linhas telefônicas comuns.

À medida em que um dispositivo celular, como um telefone portátil, move-se de uma localidade para outra, o sinal pode cair momentaneamente conforme se passa de uma área coberta por uma célula para outra área coberta por uma segunda célula. Isso é conhecido como *handoff*.

Quando um dispositivo celular move-se para dentro do alcance de uma nova célula, seu sinal é capturado por outro par formado por um transmissor e um receptor. Normalmente esta célula é a que encontra-se mais próxima do dispositivo. Entretanto, se um sinal é bloqueado, uma célula mais afastada passa a ser utilizada. [26]

As ocorrências de *handoff* não são problema para sinais de voz, cheios de vazios e pausas; contudo, no caso dos dados, se não houver protocolos ou equipamentos especiais, cada troca de célula representa uma potencial transmissão arruinada. [27]

2.4.2 Celular Analógico

Uma das formas mais simples de comunicação sem fio é o celular analógico. Projetado originalmente para comunicações de voz, o celular analógico funciona de forma muito parecida com os telefones baseados em linhas de terra – na verdade, uma chamada celular viaja a maior parte da distância através dessas linhas. A transmissão de dados via celular analógico requer um modem ligado ao serviço celular [27].

Os telefones celulares analógicos normalmente utilizam a tecnologia FDMA (*Frequency Division Multiple Access*) que divide as frequências para separar cada uma das chamadas [46].

2.4.3 CDPD (Cellular Digital Packet Data)

Um dispositivo celular CDPD segmenta os dados em pacotes que podem ser transmitidos de forma independente. Estes pacotes de dados são enviados por meio de sinais de celulares que já estão sendo utilizados para comunicações de voz; eles são inseridos na portadora do sinal durante os períodos de silêncio que acompanham a fala.

Devido à possibilidade de perda dos pacotes de dados durante um *handoff*, ou de distorção causada pela estática ou por outra interferência no sinal do celular, o CDPD receptor verifica os pacotes para ter certeza de que não apresentam problemas, e solicita ao CDPD emissor que torne a enviar os pacotes danificados ou perdidos. [26]

2.4.4 TDMA (Time Division Multiple Access)

Método de multiplexação para telefones celulares digitais, no qual é atribuído uma faixa de tempo em um canal para cada usuário.[26] TDMA divide cada frequência de portadora em um determinado número de intervalos de tempo, cada qual constituindo um circuito independente de telefone.[46] Tipicamente, até dez usuários podem compartilhar um canal. Já existe cerca de 30 milhões de usuários que utilizam celulares com esta tecnologia [37].

2.4.5 GSM (Global System for Mobile Communication)

O GSM utiliza a mesma tecnologia do TDMA onde cada frequência de portadora é dividida em um número de faixas de tempo. GSM é utilizado amplamente na Europa como principal padrão para sistemas de comunicação sem fio.[46]

2.4.6 Outros tipos de comunicação

Existe um número grande de estratégias usadas na comunicação sem fio. Elas são importantes mas não fazem parte do escopo deste trabalho, por isso serão apenas citadas: Microondas [6], Frequency-Hopping Spread Spectrum, Direct-Sequence Spread Spectrum Technology [51] [43] [48] [5], Code Division Multiple Access (CDMA) [26] [46] [6], Wideband Code Division Multiple Access (W-CDMA) [52], Infravermelho [51] [6] [5], etc.

2.5 Como uma rede sem fio funciona

Uma rede sem fio utiliza ondas eletromagnéticas para transmitir informações de um ponto a outro sem o uso de conexões físicas. Ondas de rádio são frequentemente chamadas de portadoras de rádio porque elas simplesmente executam a função de enviar energia para um receptor remoto. O dado transmitido é sobreposto na portadora do rádio de forma a permitir a extração desta informação no ponto receptor. Uma vez o dado sobreposto (ou modulado) na portadora do rádio, este sinal ocupa mais do que uma única frequência, uma vez que a frequência ou a taxa de bits da informação modulada é adicionada à portadora. Múltiplas portadoras de rádio podem existir dentro do mesmo espaço e dentro do mesmo tempo sem interferir um no outro se estas ondas forem transmitidas em diferentes

freqüências de rádio. Para extrair os dados, o rádio receptor é ajustado em uma freqüência enquanto as demais freqüências são descartadas.

Em uma configuração de rede típica, o dispositivo de transmissão/recepção, chamado de ponto de acesso, conecta-se com a rede através de uma base fixa que usa cabos tradicionais (Figura 2.1). O objetivo do ponto de acesso é receber, armazenar e transmitir os dados entre as redes sem fio e com fio. Um único ponto de acesso pode suportar um pequeno grupo de usuários e pode funcionar dentro de uma faixa que varia de uma a várias centenas de metros. O ponto de acesso, ou a antena anexada ao ponto de acesso, é normalmente colocado em um local alto, entretanto pode ser colocado em qualquer lugar onde o alcance do rádio atinja a área de cobertura desejada. Os pontos de acesso não apenas fornecem a comunicação com a rede convencional, como também intermediam o tráfego com os pontos de acesso vizinhos, num esquema de micro células com *roaming* semelhante a um sistema de telefonia celular (figura 2.1). Os usuários acessam a rede sem fio através de adaptadores de rede especiais, que disponibilizam uma interface entre o sistema operacional de rede (SOR) e a antena transmissora do sinal. A partir desta estrutura a natureza desta conexão sem fio fica transparente para o SOR. [51]

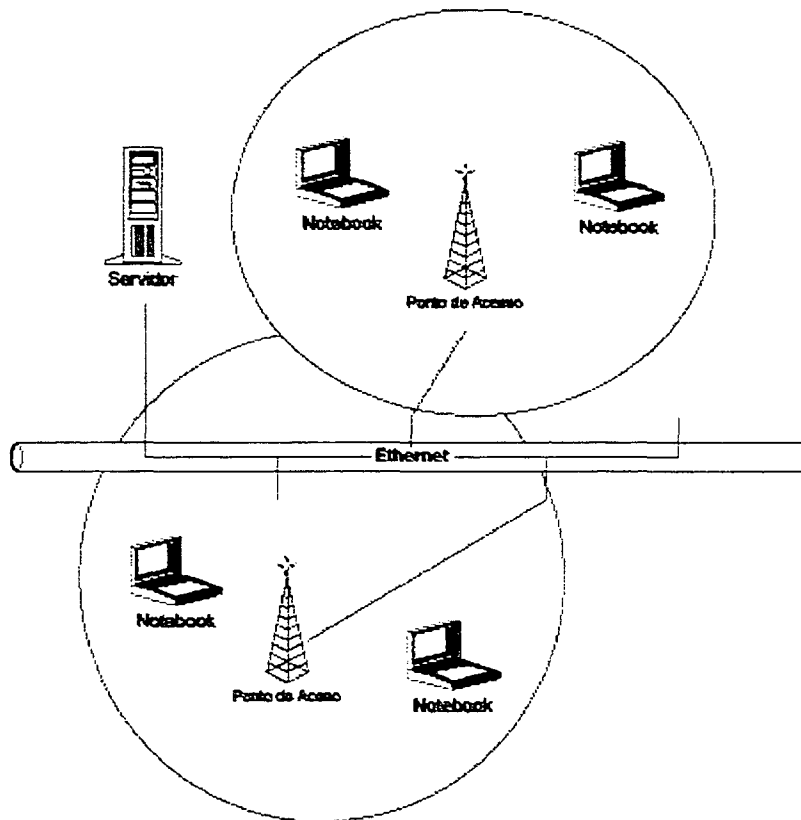


Figura 2.1 — Exemplo de um ambiente de rede sem fio usando pontos de acesso

2.6 Mercado Atual

2.6.1 Comunicação de dados via celular

Os modems atuais foram originalmente projetados para linhas telefônicas convencionais, que possuem as seguintes características:

- Comunicação "estável" após estabelecimento da ligação.
- "Pequenas" variações das características do canal de comunicação.
- Caminho/distância do sinal fixos durante a comunicação.

O sistema celular foi projetado simplesmente para manter um mínimo de qualidade de sinal de voz, apresentando:

- Variações do nível do sinal durante a ligação.
- Interrupções da comunicação, devido a troca de frequências (*hand-off*).
- Mudança do percurso do sinal durante a comunicação.
- Eco do sinal devido a reflexões ("fantasmas")

Todos estes fenômenos são devidos às características da comunicação via sinal de rádio dividida em células. Assim, fica claro que a comunicação de dados via telefonia celular representa um desafio para o modem, pois as condições são muito mais adversas do que as apresentadas pelo sistema de telefonia convencional.

O primeiro ponto que deve ser levado em consideração é a utilização de modems convencionais, usados em linhas telefônicas comuns, que somente podem ser aproveitados no sistema celular analógico, já que esses modems utilizam este modo (analógico) para comunicação. Assim, os sistemas digitais existentes atualmente (CDMA / TDMA) por enquanto não suportam a comunicação de dados.

Tendo em vista as condições adversas do canal celular, se faz necessário o uso de um protocolo especial, que trate os problemas apresentados há pouco e tenha maior tolerância às condições da linha celular. Atualmente, um protocolo muito utilizado é o MNP10, que possui uma versão aperfeiçoada, conhecida como MNP10EC. Exemplos de aplicações onde não se pode utilizar o protocolo MNP10EC, por que a outra ponta não trabalha com o protocolo, é o acesso a RENPAC e a muitos provedores Internet.

É importante notar, também, que o protocolo MNP10EC melhora sensivelmente a qualidade e estabilidade da comunicação, entretanto, em locais em que a comunicação via celular possuem muitos aparelhos dentro da mesma célula ou que possuem muitos obstáculos, o que ocorre nas capitais de São Paulo e Rio de Janeiro, mesmo com este protocolo não há garantias da qualidade na comunicação. O último aspecto a ser lembrado sobre o MNP10EC é que ele só opera para transmissões de dados, não tendo função para transmissões de Fax, assim, estas normalmente têm um índice relativamente baixo de sucesso, uma média de 30% a 50%.

Devido as limitações de banda do canal celular, a comunicação fica limitada a velocidades entre 2400bps a 4800bps. Usando compressão de dados é possível chegar a uma média de 9600bps. Assim, é importante que as aplicações utilizadas não exijam alto tráfego de dados, devendo-se ater a transmissões de arquivos pequenos.

Requisitos para Comunicação de Dados via Telefonia Celular

- Os modems de ambos os lados da comunicação (origem/resposta) devem possuir o protocolo MNP10EC.
- Os locais de comunicação devem apresentar um sinal de boa qualidade.
- Existir adaptador RJ-11 do telefone celular.

Sendo assim, este meio de comunicação é mais recomendado para aplicações específicas que possuem baixo volume de transmissão.

Existem duas tecnologias que prometem soluções para os problemas apresentados:

- A CDPD (Cellular Digital Packed Data), é um sistema de comunicação de dados operando nas velocidades de 9,6kbps a 19,2kbps para o sistema celular, necessitando a instalação de equipamentos de recepção nas estações rádio-base e a utilização de um modem especial CDPD. Algumas operadoras de celular no Brasil já disponibilizam este tipo de comunicação [32].
- O sistema PCS (Personnal Communications Services), é um novo sistema de telefonia celular digital, que prevê a comunicação de dados. [42]

2.6.2 Acesso à Web via celulares, pagers e computadores de mão

No início de 1999 a Motorola e a Cisco anunciaram investimento de US\$ 1 bilhão durante cinco anos para criar um padrão de acesso à Internet a partir de redes sem fio. Sprint, Nextel e AirTouch foram algumas das empresas que aderiram à iniciativa. Ao mesmo tempo, a Microsoft firmou um acordo com a British Telecom para desenvolver um navegador destinado a telefones celulares. As duas notícias sinalizaram a tendência que a Internet começou a experimentar em 1999: *wireless*. Ler e-mails ou buscar informações na *Web* não são mais operações exclusivas dos micros de mesa, isso pode ser feito usando o celular através dos serviços da America on Line [30], MSNBC ou alguns sites nacionais como o PageMe [31]. Neste segmento o protocolo WAP (*Wireless Application Protocol*) está sendo amplamente adotado.

A Web sem fio através do celular ainda é lenta, funciona a 14,4 Kbps contra os 56K dos modems nos PCs. O tamanho dos visores dos telefones móveis varia de duas a três polegadas contra os de 15 ou 17 polegadas nas mesas do escritório. Apesar disso, segundo a (IDC) *International Data Corporation*, dos 64 milhões de proprietários americanos de

celulares, 15% já utilizam seus equipamentos para o tráfego de dados. Em 2002, quando os Estados Unidos tiverem cerca de 108 milhões de telefones celulares, 70% deles serão usados para buscar informações na Internet. A IDC prevê que as alterações percentuais serão mais ou menos semelhantes para o resto do mundo, para o Brasil a previsão da Anatel é de 30 milhões de celulares em 2003. Considerando todos os aparelhos sem fio que acessam a Web a IDC assinalou 7,4 milhões de usuários em 1999 com a previsão de que este número chegue a 61,5 milhões em 2003 nos EUA [34].

O grande problema é como transportar os serviços já criados para computador à tela dos telefones celulares. Com objetivo de facilitar este processo estão sendo criados padrões como o *Short Message Services (SMS)*, *AnyWeb* e *Wireless Application Protocol (WAP)*. O WAP recebeu apoio da Ericsson, Nokia e Motorola, contando com a adesão de gigantes do software como IBM e Microsoft. O WAP reúne um ambiente de aplicações e um conjunto de protocolos de comunicação para dispositivos sem fio. [39] Os sinais da disseminação desta tecnologia já aparecem no Brasil. No início de 2000 a Nokia e a Motorola lançaram telefones celulares que utilizam a tecnologia WAP. Além desta tecnologia outra tendência da utilização da tecnologia sem fio também está presente nestes novos celulares, que através de infravermelho permitem a sincronização e comunicação dos dados entre telefones, PCs e impressoras [36].

A partir de abril de 2000, entra na Web em versão definitiva o site PageMe (www.pageme.com.br), um portal voltado à tecnologia sem fio que pretende ser o ponto de partida para o acesso à Web via celulares, *pagers* e computadores de mão.

O site está em fase experimental desde Dezembro. Inicialmente este site utiliza a tecnologia SMS para enviar pequenas mensagens da Web, como e-mails, dicas culturais e cotações, para os aparelhos móveis. Com a chegada da tecnologia WAP serviços mais amplos e de maior valor agregado vão ser possíveis, como *home banking*, reservas de vôos, diárias de hotéis e investimentos online. [31]

2.6.3 Redes locais sem fio

Considerando todas as vantagens existentes na utilização de redes sem fio, a explosão de vendas deste mercado, previsto para a década de 90, nunca realmente aconteceu. A razão para este crescimento lento pode ser justificada por dois motivos: preços altos e a falta de padronização no mercado. Estes fatores criam um mercado desfavorável, porque além do alto custo de implantação destes sistemas, a única alternativa era a adoção de soluções de hardware e software proprietárias.

Para resolver este problema do mercado, em Maio 1991 um grupo de pessoas liderados por Victor Hayes, submeteu um projeto ao IEEE (*Institute for Electrical and Electronic Engineers*) para a criação do grupo de trabalho 802.11. Este trabalho durou quase toda a década de 90 gerando em Junho de 1997 o documento definindo padrões para redes locais sem fio (WLANs).

A existência deste padrão finalmente assegura aos usuários a possibilidade de comprar produtos interoperáveis de vários fornecedores diferentes para montar ou expandir a rede. Isso faz com que a competição aumente e conseqüentemente os custos diminuam.

2.7 Futuro do mercado sem fio

A falta de uma cobertura universal é o grande fator que está retardando o uso mais difundido de serviços de dados sem fio. Muito em breve, graças às plataformas digitais, que exploram meios de transmissão sem fio, padronizados e abertos, as pessoas passarão a utilizar um único meio, identificado por um só número, para enviar e receber mensagens de dados, voz e imagem. Isso será possível devido à convergência de três diferentes indústrias: telecomunicações, informática e televisão. De maneira a incentivar o desenvolvimento dessa indústria multimídia, em 1994 criou-se a GII (*Global Information Infrastructure*). A GII compete promover a implantação da superinfovia da comunicação baseada nos seguintes princípios:

- todos os serviços serão concentrados em uma rede digital única;
- a capacidade de transmissão será abundante; e
- os serviços oferecidos serão pessoais.

Essa rede digital será diversificada e fragmentada, com elevado grau de competição (fibra/sem fio e satélite/cabo), sustentando-se em plataformas tecnológicas abertas. A proposta da IMT-2000 é fornecer acesso *wireless* à infra-estrutura global de telecomunicações, mediante unificação dos padrões adotados em sistemas móveis celulares em uma única infra-estrutura de rádio, provendo, ao mesmo tempo, serviços de voz, dados e banda larga para a multimídia com *roaming* global. Entre os diversos padrões tecnológicos de arquitetura de rede e interface aérea, o que melhor atende os requisitos do IMT-2000 é o W-CDMA (*Wideband Code Division Multiple Access*).

A fusão das tendências GII e IMT-2000 dará origem a uma rede mundial sem fio digital de alta capacidade, via sistema móvel celular. E independentemente do meio utilizado para transportar o sinal – via fibra, satélite ou infra-estrutura sem fio.

Para as residências o crescimento do mercado sem fio acontecerá na construção das infra-estruturas a serem utilizadas pelas novas operadoras como, por exemplo, na telefonia fixa, através da utilização da tecnologia WLL (*Wireless Local Loop*) lançada em Dezembro de 1999 pela operadora Vésper em São Paulo [35]. [29]

Já existe o que está sendo chamado de celular de terceira geração, com alguns aparelhos lançados na Europa. Este celular permite navegar na Web, consultar a sua conta bancária, receber informações e gravar músicas no padrão MP3. A característica fundamental do celular de terceira geração é a sua maior capacidade de transmissão de dados. Em vez de acessar a Web à velocidade de 19,6 kbps, esta terceira geração trabalha a 384 kbps e, nos sistemas que utilizam WLL (*Wireless Local Loop*), 2 megabits/segundo.

No Brasil, a indústria aguarda as regras para a entrada de um novo competidor no segmento, conhecida como banda C. O assunto sobre a chegada do PCS (Personal Communications System) tem sido discutido pela indústria e eventuais grupos interessados em explorar o serviço. A Agência Nacional de Telecomunicações (Anatel) deverá definir ainda em 2000 a estratégia não apenas para a banda C, mas para a terceira geração de telefonia celular. Uma questão técnica já foi levantada pela indústria e pelo CDG (CDMA Development Group). Começa pela frequência de operação da banda C. A agência tem duas opções: 1,8 GHz ou 1,9 GHz. O PCS opera em 1,9 GHz nos Estados Unidos, Canadá, Chile, Argentina, México, Peru, Uruguai e Venezuela. Na Europa, essa faixa será utilizada

pelos sistemas de terceira geração. E qual será o modelo do Brasil? Empresários defendem que o Brasil deveria seguir o caminho dos demais países, ou seja, 1,9 GHz. Alegam que o *roaming* do serviço poderia ser feito sem maiores complicações e a migração da segunda para a terceira geração não seria tão cara. A indústria já desenvolveu infra-estrutura e aparelhos na faixa 1,9 GHz nos três padrões digitais: TDMA, CDMA e GSM. Isto significa que é possível aumentar a escala de produção e baratear o custo para o usuário final já que a indústria tem equipamentos e aparelhos *dual band*. No caso da frequência de 1,8 GHz, a opção de tecnologia disponível é o GSM, que não é usada no Brasil. [33]

A tendência é de que as telecomunicações adotem um sistema pessoal sem fio, com ampla gama de frequências alocadas, capaz de prover um conjunto de serviços de comunicação com elevada capacidade de transmissão e aplicações multimídia. Um conjunto de serviços que já se convencionou chamar de infocomunicação sem fio.

3. PESQUISAS NA ÁREA

3.1 Introdução

Os sistemas celulares modernos utilizam uma estação base que cobre uma região geográfica específica, chamada célula. Os problemas aparecem na decisão de quando trocar de uma célula para outra, na manipulação da interferência entre as células e na localização do usuário. Mesmo com estes problemas complexos, a comunicação sem fio tem um futuro lucrativo e garantido. Características como a liberdade de comunicar/mover e a capacidade de levar a comunicação para localidades remotas fazem com que esta tecnologia seja parte importante da nossa sociedade.

3.2 O Sistema ALOHA da Universidade do Haváí

Foi o primeiro sistema de computadores a empregar a técnica de radiodifusão no lugar de cabos ponto a ponto. Na década de 70, quando o projeto foi implantado, as linhas telefônicas disponíveis eram caras e pouco confiáveis. Havia a necessidade de interligação de subredes da universidade, espalhadas pelas ilhas, ao Centro de Computação principal.

A comunicação foi realizada através da instalação, em cada estação, de um pequeno transmissor / receptor de rádio FM, com um alcance suficiente para comunicar-se com o transmissor / receptor do Centro de Computação.

O projeto foi realizado de uma forma que não existia comunicação direta entre estações, apenas de uma estação para o Centro de Computação e deste para uma estação. Foram utilizadas duas faixas de frequência: uma em 407.305MHz para o tráfego no sentido

Centro-Estação, e outra em 413.475MHz para o tráfego no sentido contrário. A transmissão foi feita em 9600bps.

A figura 3.1 apresenta os elementos básicos do sistema ALOHA.

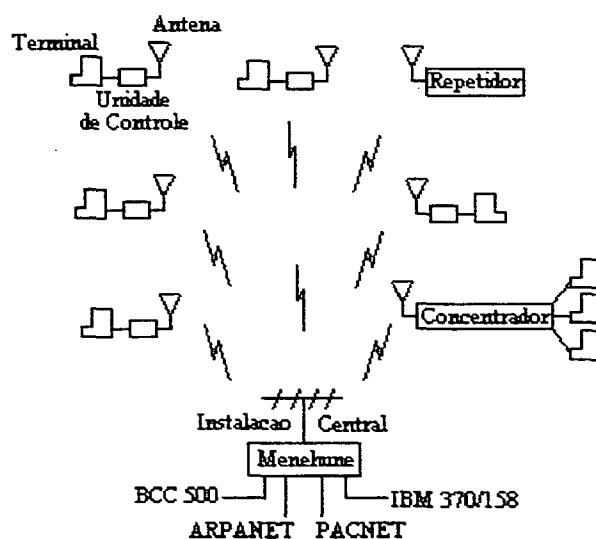


Figura 3.1 — Elementos básicos do sistema ALOHA

Na instalação central encontra-se um computador chamado *Memhune* que manipula todos os dados recebidos ou transmitidos. O *Memhune* está conectado ainda a outros dois computadores (BCC 500 e IBM 370/158) e as redes ARPANET e PACNET. Cada estação possui uma unidade de controle que armazena dados e faz as retransmissões.

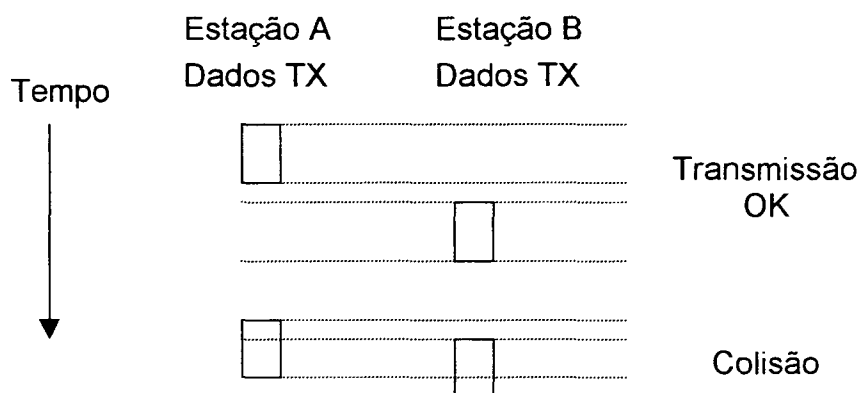


Figura 3.2 — Colisão na Transmissão do Sistema ALOHA

O método de acesso utilizado na rede ALOHA é simples. Cada terminal só pode ouvir o canal de transmissão do computador para o terminal, não tendo condições de saber se o outro canal está sendo utilizado por outro terminal ou não. Quando um terminal tem um pacote para transmitir, este pacote é transmitido, independentemente de o canal estar sendo utilizado ou não. A técnica de detecção de colisão é realizada pelo disparo do *timer* na transmissão da mensagem. Se um pacote de reconhecimento não tiver chegado ao final da temporização, o pacote original deve ser retransmitido. O intervalo de temporização é aleatório de forma a reduzir a probabilidade de nova colisão de pacotes.

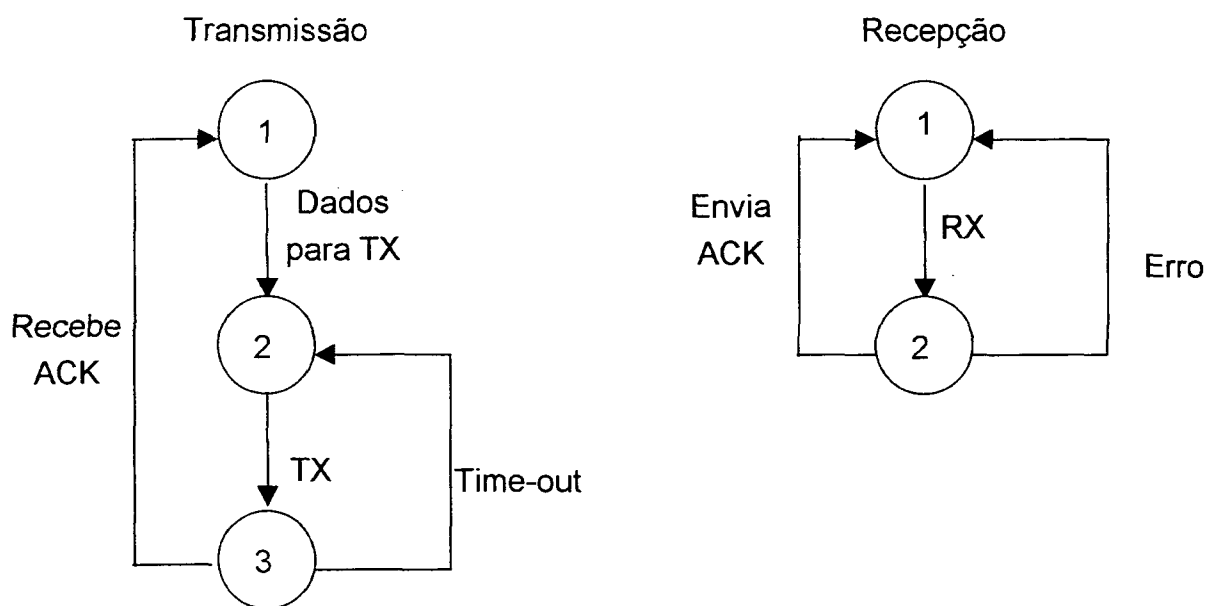


Figura 3.3 — Diagrama de Transmissão e Recepção do sistema ALOHA

3.3 Áreas de Pesquisa na Computação sem fio

A comunicação sem fio é muito mais difícil de conseguir do que a comunicação com fio porque o ambiente tem uma grande influência no sinal da comunicação, exemplo: bloqueios no sinal transmitido, ocorrência de ruídos e ecos na transmissão. Por este motivo,

quando comparada com comunicação que usa fios, as transmissões sem fio tem menor velocidade de transmissão e uma maior frequência de erros. Além disso, a comunicação sem fio tem outros problemas quando utilizada em equipamentos móveis. Estes equipamentos podem sair da área de abrangência dos transmissores, ou entrar em áreas de alta interferência, gerando a perda da comunicação ou a degradação do sinal. Diferentemente das redes com fio, o número de usuários conectados a uma rede sem fio pode variar dinamicamente dentro de uma célula, como por exemplo a concentração de muitos usuários com equipamentos sem fio em um congresso ou uma feira. Isso pode gerar picos que sobrecarregam a rede. A possibilidade do usuário se movimentar enquanto está conectado à rede aumenta a complexidade de algumas informações. Alguns dados, considerados fixos em redes com fio, como endereço na rede ou nome do servidor, passam a ser informações dinâmicas, obrigando a rede a modificar ou procurar por novas configurações enquanto o usuário se movimenta, por exemplo, um novo endereço de rede ou o nome do servidor mais próximo.

Existem ainda muitos problemas a serem resolvidos na área de comunicação sem fio, como por exemplo: ocorrência de frequentes desconexões, a velocidade na comunicação, grande variação na capacidade de transmissão, redes heterogêneas e os riscos de segurança. Além destes problemas, existem outros relacionados diretamente aos equipamentos que se beneficiam deste tipo de comunicação, como por exemplo o problema da mobilidade (migração de endereço e informações dependentes de localização) e a portabilidade destes equipamentos (*Low Power* e risco dos dados). [17]

3.3.1 Desconexão

Hoje os computadores dependem muito de uma conexão a rede para funcionar com todas as suas características, como o correio eletrônico, a Internet e acesso ao banco de dados. Como as perdas de conexão são mais freqüentes quando é utilizado redes sem fio, a dependência do próprio sistema operacional à disponibilidade do acesso a rede se torna um problema para estes computadores.

Uma alternativa para trabalhar com este problema é gastar os recursos da rede na tentativa de prevenir as desconexões, ou ao invés disso considerar que estes problemas são inevitáveis e criar sistemas que conseguem conviver com este tipo de problema. Quanto mais autônomo o sistema, mais tolerante este sistema será a ocorrência de desconexões. Por exemplo, uma aplicação pode reduzir a necessidade do uso de rede se executar a maior quantidade de tarefas localmente do que distribuir a aplicação na rede. Em um ambiente onde ocorrem desconexões freqüentes, é importante que o computador tenha autonomia e consiga trabalhar fora da conexão de rede.

Em sistemas sem fio muitas vezes é possível que problemas como a latência na transmissão das mensagens ou rápidas desconexões sejam minimizados através da utilização de operações assíncronas. Diferente do funcionamento das chamadas remotas síncronas aos procedimentos, onde o sistema aguarda pela resposta antes de continuar o processamento, a utilização de operações assíncronas permitirão ao sistema executar uma série de chamadas antes de aguardar por uma resposta. Da mesma forma a utilização de técnicas como *prefetching* e *lazy write* [10] permitem que o sistema continue funcionando, ou o usuário continue trabalhando, sem a necessidade de que todas as operações

dependentes dos recursos da rede tenham que acontecer em tempo real. A desvantagem do uso destas técnicas é a possibilidade de mascarar eventuais problemas na rede. Por isso, o sistema não pode desconsiderar totalmente a possibilidade de problemas na rede. Quando ocorrer uma falha o sistema pode informar ao usuário quais os serviços estarão indisponíveis até que o acesso a rede seja recuperado.

3.3.2 Largura de Banda

Os equipamentos/sistemas que utilizam redes sem fio precisam considerar em seu projeto a largura de banda utilizada durante o seu funcionamento, procurando reduzir ao máximo a necessidade de transmissão de dados na rede. Diferente dos sistemas com fio, a velocidade da maior parte das redes sem fio, como por exemplo 1Mbps para comunicação infravermelha ou 2Mbps para comunicação por rádio, é muito inferior a capacidade de uma rede com fio que, por exemplo, em uma rede *Ethernet* pode ser de 10 ou 100Mbps.

Se for considerada uma rede sem fio utilizada por usuários móveis, outro fator que tem influência na capacidade de transmissão da rede está relacionado a quantidade de pessoas que a rede precisa atender em uma determinada área. Redes para atender usuários móveis utilizam a mesma estratégia da telefonia celular, onde a área de abrangência da rede é dividida entre vários transmissores cada um sendo responsável por atender um usuário dentro de uma célula. Se uma quantidade muito grande de usuários se concentrarem dentro de uma mesma célula, estará sendo dividida entre eles a capacidade de transmissão da célula.

Uma forma de aumentar a capacidade da rede é através da inclusão de mais células. Isso pode ser feito através da sobreposição de células com diferentes frequências ou na

diminuição da potência de transmissão de uma célula, reduzindo a sua área de abrangência, porém permitindo que mais células possam existir dentro de uma determinada área.

A escalabilidade da primeira alternativa (múltiplas frequências) é restrita pois o espectro eletromagnético disponível para uso público não é muito grande.

Já a segunda alternativa (redução da potência) é a mais utilizada, porque os custos associados a esta solução se restringem apenas ao aumento do número de transmissores para atender uma área. Além disso existe uma relação entre largura de banda e área de cobertura, onde os transmissores/receptores que fazem a cobertura de uma área menor podem atingir uma maior largura de banda.

Para diminuir o problema relacionado a largura de banda é possível utilizar estratégias de software, como por exemplo a compressão dos dados antes de transmiti-los. Como a utilização da rede possui muitos momentos de pico, onde a rede oscila entre instantes de ociosidade e de demanda superior a capacidade da rede, pode-se minimizar este tipo de problema utilizando as técnicas como *prefetching* e *lazy-write*, mencionado no item desconexão, já que o problema de picos de transmissão na rede se assemelham ao problema da desconexão.

É possível combinar as soluções de hardware/software apresentadas a outras medidas, ou políticas, durante a utilização da rede. Por exemplo, operações de *backup* podem acontecer apenas durante os períodos de ociosidade da rede ou o envio lento das mensagens de correio eletrônico aos seus usuários.

3.3.3 Grande Variabilidade na Largura da Banda

Muitas vezes os equipamentos que utilizam comunicação sem fio também permitem ao usuário se conectar a uma rede com fio. Esta possibilidade tem consequências para as aplicações pois elas poderão ser utilizadas em situações de maior (rede com fio) e menor (rede sem fio) largura de banda. Para tratar deste problema, as aplicações podem utilizar três técnicas diferentes. A primeira consiste em permitir a utilização da aplicação apenas quando existe largura de banda suficiente. A segunda é o desenvolvimento da aplicação considerando apenas a existência de pouca largura de banda e, se uma maior largura de banda estiver disponível a aplicação desconsidera a sua existência. Por fim a terceira alternativa é fazer com que a aplicação tenha níveis de funcionamento que são ativados automaticamente reduzindo ou aumentando o número de características/detalhes ao usuário conforme a largura de banda disponível.

3.3.4 Redes Heterogêneas

Redes sem fio para usuários móveis precisam tratar de alguma forma o problema da existência de diferentes tipos de rede, que podem variar em qualidade, tipo de interface ou protocolo utilizado. Por exemplo, a qualidade de acesso a rede sem fio pode ser melhor em uma sala de reuniões do que o acesso disponível enquanto o usuário está se movimentando na rua. Durante a sua movimentação o usuário poderá estar em uma área onde é possível acesso a mais de um transmissor/receptor operando em diferentes frequências. É possível que enquanto o equipamento sem fio esteja fisicamente conectado a uma rede, ele continue tendo acesso a uma rede sem fio. Outra possibilidade ocorre quando o usuário está utilizando o seu equipamento dentro de uma área fechada que disponibiliza aos clientes

móveis uma interface infravermelho. Se este mesmo usuário continuar necessitando de acesso a rede enquanto estiver em um ambiente aberto, um outro tipo de interface deverá ser disponibilizada pois a comunicação por infravermelho não funciona corretamente quando existe a interferência da luz do sol. Agora, mesmo que o usuário esteja utilizando rádio frequência pode ser necessário uma mudança de protocolo de comunicação quando passar da cobertura celular em uma cidade, para uma cobertura por satélite em um país. Esta mistura de tecnologias e protocolos faz com que as redes que suportam usuários móveis sejam mais complexas que as redes tradicionais.

3.3.5 Riscos de Segurança

Segurança é um aspecto importante em qualquer tipo de rede e normalmente é tratada em dois níveis: através de políticas de acesso e segurança na transmissão das informações.

- **Políticas de acesso:** direitos de acesso, domínios, grupos de usuários, entre outras.
- **Segurança na transmissão das informações:** evitar que a transmissão seja capturadas e visualizadas por equipamentos/pessoas não autorizadas.

Estes dois aspectos são ainda mais difíceis de serem tratados por redes sem fio. Por exemplo, as políticas de acesso devem permitir usuários móveis que estão fora do seu domínio (*untrusted users*), por exemplo em um centro de convenções, acessem impressoras próximas, porém não tenham acesso a recursos que apenas os funcionários daquele local têm direito. Quanto a transmissão dos dados nas redes sem fio existe o agravante da facilidade de acesso as informações transmitidas, principalmente em redes sem fio de grande abrangência. Como não é possível impedir que pessoas não autorizadas tenham

acesso as informações transmitidas, é necessário trabalhar com alternativas que impeçam que as informações transmitidas possam ser compreendidas pelos usuários que não tenham direito de acesso a esta rede. Isso pode ser feito utilizando técnicas de criptografia em software ou em hardware, como por exemplo o uso de transmissões usando a técnica de espalhamento espectral (*spread spectrum*).

3.3.6 Migração de Endereço

Enquanto as pessoas se movimentam, os seus equipamentos utilizam diferentes pontos de acesso. As redes atuais não estão preparadas para a mudança dinâmica no endereços das estações. No protocolo Internet (IP) os nome dos computadores na rede estão associados aos respectivos números IP. Movimentar um equipamento para um outro local normalmente requer a intervenção humana para a modificação do seu número IP. Para ser possível se comunicar com um equipamento móvel é necessário saber qual é o seu endereço atual. Existem quatro mecanismos básicos para determinar o endereço atual de um equipamento móvel: difusão (*broadcast*), serviços centralizados, *home basis* e *forwarding pointers*. Estes mecanismos são a base da proposta para a existência de IP móveis.

3.3.7 Difusão Seletiva

No método da difusão (*broadcast*), uma mensagem é enviada a todas as células da rede, perguntando ao equipamento móvel qual o seu endereço de rede atual. Esta técnica se torna muito cara se o seu uso for freqüente em uma grande rede (muitas células), entretanto se a rede sabe que provavelmente o usuário está localizado dentro de um pequeno conjunto de células desta rede, a difusão pode ser feita de forma seletiva, considerando apenas este conjunto de células. Desta forma a técnica da difusão passa a ser viável.

3.3.8 Serviços Centrais

Neste método o endereço atual de uma estação é mantido em um banco de dados logicamente centralizado. Cada vez que o equipamento muda o seu endereço ele é enviado a este banco de dados. Mesmo sendo este banco de dados logicamente centralizado as técnicas de distribuição e replicação de banco de dados podem ser utilizadas para melhorar a disponibilidade e o tempo de resposta para o acesso a estas informações.

3.3.9 Home Bases

Nesta técnica existe diversos servidores, cada um responsável pelo controle de uma estação. A localização da estação é conhecida por apenas um destes servidores. A distribuição da responsabilidade de controle das estações gera o problema da disponibilidade. Se um destes servidores estiver indisponível não será possível encontrar as estações que estão sob seu controle.

3.3.10 Transferência de Ponteiros

Neste método a cada vez que uma estação mude o seu endereço , uma cópia do seu novo endereço é colocado no endereço antigo. Cada mensagem enviada segue portanto uma cadeia de endereços até chegar na estação desejada. De forma a evitar a o tempo gasto no roteamento de mensagens por longas cadeias de endereços, os ponteiros podem ser atualizados para refletir endereços mais recentes. Esta técnica requer a existência de entidades ativas nos endereços utilizados pelas estações móveis, pois estas entidades serão responsáveis pelo envio da mensagem para o endereço atualmente ocupado pela estação.

3.3.11 Informações Dependentes de Localização

Uma rede com fio não espera que os seus equipamentos troquem de lugar frequentemente. Por este motivo a configuração das informações que dependem da sua localização, por exemplo o nome do servidor ou das impressoras próximas, são estáticos. Um desafio para as redes sem fio que permitem que os seus usuários se movimentem dentro de uma grande rede, é conseguir obter de forma inteligente este tipo de configuração de acordo com o local que o equipamento se encontra.

Além deste problema de configuração dinâmica, os usuários móveis precisam de outras informações que estão relacionadas à sua localização, como por exemplo onde está o posto de gasolina mais próximo.

3.3.12 Low Power

A bateria é um dos componentes existentes nos equipamentos móveis responsável em grande parte por seu peso e tamanho. Enquanto a redução do peso da bateria é importante, essa redução não pode influenciar a autonomia do aparelho. O balanço entre o tamanho, peso e durabilidade da bateria tem grande influência sobre a utilidade do equipamento móvel.

3.3.13 Risco para os Dados

Utilizar equipamentos portáteis aumenta o risco quanto à segurança das informações como os acessos não autorizados ou até mesmo os acidentes com estes equipamentos. Estes riscos podem ser diminuídos através da redução da quantidade de informações essenciais guardadas dentro destes equipamentos, ou então que estas informações sejam

automaticamente copiadas em forma de *backup* em servidores desta rede. Sobre a segurança contra acessos não autorizados a estas informações, é importante que o equipamento somente guarde dados criptografados em seu disco rígido ou cartão de memória.

3.3.14 A Confiabilidade na Transmissão

A transmissão de pacotes de uma rede sem fio não é uma operação confiável. Esta menor confiabilidade cria alguns problemas de performance na camada de transporte. De forma a transformar o meio “mais confiável” para os protocolos da camada de transporte, a forma usual de tratar este problema é incluir retransmissões no MAC de acordo com a proposta do padrão IEEE para redes sem fio 802.11.

Para pacotes do tipo *broadcast* e *multicast*, o problema é ainda pior. A camada de transporte não pode incluir qualquer mensagem de reconhecimento devido ao número indefinido de receptores e as diferentes condições na recepção. Por isso técnicas alternativas precisam ser utilizadas para o reconhecimento e a retransmissão deste tipo de mensagens pelo MAC. [20][19]

3.4 Especificação Bluetooth

Bluetooth é uma especificação que define como PCs, celulares, e outros equipamentos portáteis podem trocar informações entre si. Este padrão está baseado no conceito de WPAN (*Wireless Personal Area Network*) que por sua vez está baseado no conceito de POS (*Personal Operating Space*). POS compreende uma área de aproximadamente 10 metros ao redor de uma pessoa. [23]

Com o objetivo de estudar a necessidade de padrões complementares voltados a conectividade de dispositivos de baixo consumo que estão dentro da área abrangida pela POS, o IEEE 802.11 montou um grupo de estudo WPAN. O resultado deste estudo foi a criação, em Março de 1999, de um novo grupo de trabalho chamado IEEE 802.15. [21]

Os estudos para a aplicação desta tecnologia de comunicação de baixo alcance e baixo custo, que são propostas pelo Bluetooth e estão de acordo com o IEEE 802.15, incluem:

- Criação de uma rede sem fio ligando os equipamentos disponíveis no escritório, como PCs, *Notebooks*, celulares, *pagers* e *palmtops*. Todos com os seus dados de agenda, compromissos, mensagens e arquivos sincronizados.
- Acesso a informações pessoais (como correio eletrônico) a partir de pontos de acesso disponíveis em aeroportos ou centros de convenções.
- Compartilhamento de periféricos, como por exemplo a utilização de uma impressora pública disponível em um centro de convenções, ou dentro de uma área de exposição de uma feira.
- Ligação de eletrodomésticos. Exemplo: o forno de microondas utiliza a televisão para avisar que a refeição está pronta.

4. MODELO PROPOSTO

4.1 Introdução

Para a apresentação de uma proposta que viabilize o desenvolvimento de uma rede sem fio é preciso definir o objetivo a ser alcançado e quais são as restrições impostas à solução.

- **Objetivo a ser alcançado:** viabilização técnica e prática da comunicação sem fio entre microcomputadores que seja transparente ao desenvolvedor.
- **Restrições:** para a montagem do protótipo estão disponíveis dois computadores com Windows instalado, dois rádios TEKK 2W 450 a 470 MHz e duas placas serial padrão Baycom (BAY96_V2).

Para alcançar o objetivo é necessário a existência de um modelo de referência que apresente um caminho viável para a solução do problema. Foi utilizado como referência o modelo OSI e o padrão IEEE 802. A proposta de rede sem fio nesta dissertação é para a plataforma Windows, por este motivo o modelo de referência precisa ser confrontado com modelo de rede do Windows. Esta comparação terá como resultado a arquitetura da solução.

4.2 Modelo de referência

As redes sem fio realizam funções semelhantes as redes que usam cabos, como *Ethernet* ou *Token Ring*. Para permitir a transferência de informações entre os computadores, de forma geral as redes necessitam dos seguintes componentes:

- Meio que disponibilize o canal para a passagem dos dados.

- Técnicas de acesso a este meio que otimizem o compartilhamento deste canal pelas estações.
- Mecanismos de sincronização e controle de erros que assegurem a transmissão correta dos dados.
- Mecanismos de roteamento que transferem os dados da estação de origem para a estação destino desejada.
- As aplicações do usuário que realizam a troca ou o compartilhamento de dados através da rede.

Existe um grande número de alternativas para a implementação dos componentes que foram listados, como por exemplo vários meios diferentes de transmissão, técnicas diferentes para o controle/compartilhamento do canal, tratamentos de erros, protocolos e aplicativos de usuário. É tarefa do sistema operacional gerenciar todas estas alternativas e oferecer ao desenvolvedor do sistema uma interface única e consistente, propiciando uma independência ao dispositivo.

A transparência no acesso a rede sem fio terá sido atingida quando for possível ao programador, durante o desenvolvimento do seu sistema, acessar os recursos disponibilizados pela rede sem fio sem utilizar comandos ou bibliotecas especiais, apenas as chamadas disponíveis na API (*Application Program Interface*) do sistema operacional. Esta idéia está apresentada na figura 4.1.



Figura 4.1 — Usando o Sistema Operacional para disponibilizar acesso transparente a rede sem fio

Para que seja possível incorporar a rede sem fio no sistema operacional de forma transparente ao programador, é necessário identificar na estrutura deste sistema operacional quais são os componentes responsáveis pela independência do dispositivo de rede [11]. Existe padrões internacionais que definem como a camada de rede do sistema operacional deve ser organizada. É apresentado nos itens a seguir o modelo OSI (ISO) e o padrão do IEEE para redes locais.

4.2.1 O modelo OSI

Em 1984 a ISO desenvolveu um modelo de referência chamado OSI (*Open System Interconnection*) que descreve o fluxo de dados entre as aplicações do usuário e a rede. Este modelo é o mais conhecido e utilizado para descrever os ambientes de rede [2], [4], [3] e [9].

Camadas

Usuário A		Usuário B
7 – Aplicação	← Protocolo de Aplicação →	Aplicação
6 – Apresentação	← Protocolo de Apresentação →	Apresentação
5 – Sessão	← Protocolo de Sessão →	Sessão
4 – Transporte	← Protocolo de Transporte →	Transporte
3 – Rede	← Protocolo de Rede →	Rede
2 – Enlace	← Protocolo de Enlace →	Enlace
1 – Físico	← Protocolo de Físico →	Físico

4.2.2 O Padrão IEEE 802

Em Fevereiro de 1980 foi instituído o comitê IEEE 802 com o objetivo de elaborar padrões para redes locais de computadores. Este modelo está de acordo com a maioria das camadas representadas no modelo OSI, a diferença é que o padrão IEEE 802 preferiu detalhar melhor a camada de enlace, definindo uma arquitetura em 3 camadas. Estas camadas correspondem aos níveis 1 e 2 do modelo OSI [2], [4], [3] e [9].

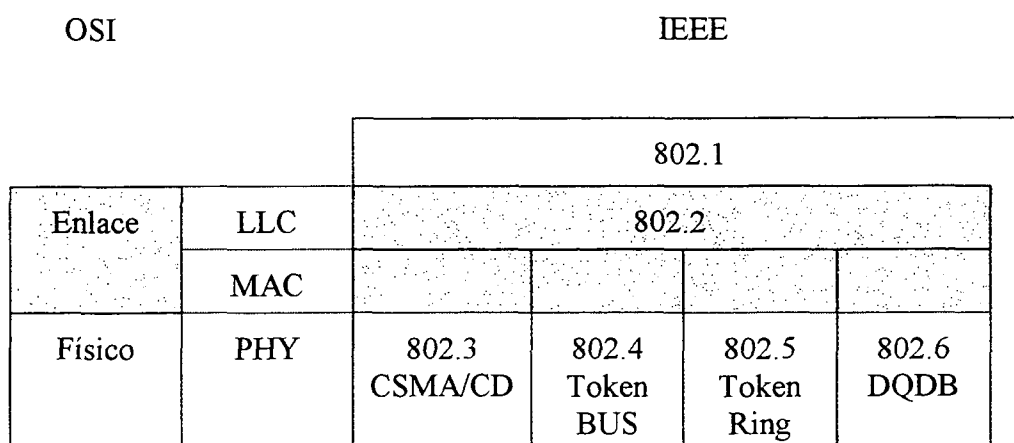


Figura 4.2 — Modelo OSI x modelo IEEE

O projeto IEEE 802 dividiu a camada de Enlace em duas subcamadas: LLC (*Logical Link Control*) e MAC (*Media Access Control*). O padrão IEEE 802.1 é um documento que descreve o relacionamento entre os diversos padrões IEEE 802 e o relacionamento deles com o modelo de referência OSI. O IEEE 802.2 descreve a subcamada superior do nível de enlace, responsável pela realização das funções de multiplexação, controle de erro e de fluxo no enlace para todos os tipos de LANs do padrão IEEE 802.

- **IEEE 802.3:** define o padrão para redes em barra usando CSMA/CD como método de acesso, como redes *Ethernet*. Aceita velocidades de 10, 100 e 1000 Mbps e meios como par trançado, cabo coaxial e fibra ótica.

- **IEEE 802.4:** define o padrão para redes em barra usando passagem de permissão como método de acesso.

- **IEEE 802.5:** define o padrão para redes em anel utilizando passagem de permissão como método de acesso.

- **IEEE 802.6:** define o padrão para redes em barra utilizando o DQDB

(*Distributed Queue Dual Bus*) como método de acesso.

4.2.3 O padrão IEEE 802.11

Com a proliferação das redes sem fio uma nova especificação foi acrescentada ao padrão IEEE 802.

	802.1				
LLC	802.2				
MAC					
PHY	802.3 CSMA/CD	802.4 Token BUS	802.5 Token Ring	802.6 DQDB	802.11 Wireless

Figura 4.3 — Padrão 802.11 acrescentado ao IEEE

Este padrão abrange uma variedade de meios físicos, incluindo FHSS (*Frequency Hopping Spread Spectrum*), DSSS (*Direct Sequence Spread Spectrum*) e infravermelho para velocidades de até 2 Mbps.

LLC		
MAC		
Frequency Hopping PHY	Direct Sequence PHY	Infrared Light PHY

Figura 4.4 — Meios físicos abrangidos pelo padrão 802.11

4.2.4 IEEE 802.11 Wireless Local Area Network

O IEEE 802.11 especifica três camadas físicas (PHY) e apenas uma subcamada MAC (*Medium Access Control*). Como apresentado a seguir, o padrão provê duas

especificações de camadas físicas com opção para rádio, operando na faixa de 2.400 a 2.483,5 MHz, que depende da regulamentação de cada país, e uma especificação com opção para infravermelho.

- **Frequency Hopping Spread Spectrum Radio PHY:** Esta camada fornece operação 1 Mbps, com 2 Mbps opcional. A versão de 1 Mbps utiliza 2 níveis da modulação GFSK (*Gaussian Frequency Shift Keying*), e a de 2 Mbps utiliza 4 níveis da mesma modulação;

- **Direct Sequence Spread Spectrum Radio PHY:** Esta camada provê operação em ambas as velocidades (1 e 2 Mbps). A versão de 1 Mbps utiliza da modulação DBPSK (*Differential Binary Phase Shift Keying*), enquanto que a de 2 Mbps usa modulação QDBPSK (*Differential Quadrature Phase Shift Keying*);

- **Infrared PHY:** Esta camada fornece operação 1 Mbps, com 2 Mbps opcional. A versão de 1 Mbps usa modulação 16-PPM (*Pulse Position Modulation* com 16 posições), e a versão de 2 Mbps utiliza modulação 4-PPM.

No lado da estação, a subcamada MAC fornece os seguintes serviços: autenticação, privacidade e transmissão da MADU (*MAC Sublayer Data Unit*), e, no lado do sistema de distribuição, a associação, desassociação, distribuição, integração e reassociação. As estações podem operar em duas situações distintas:

- **Configuração Independente:** Cada estação se comunica diretamente entre si, sem a necessidade de instalação de infraestrutura. A operação dessa rede é fácil, mas a desvantagem é que a área de cobertura é limitada. Estações com essa configuração estão no serviço BSS (*Basic Service Set*);

- **Configuração de Infra-estrutura:** Cada estação se comunica diretamente com o ponto de acesso que faz parte do sistema de distribuição. Um ponto de acesso serve as estações em um BSS e o conjunto de BBS é chamado de ESS (*Extended Service Set*).

Além destes serviços, o padrão IEEE 802 ainda oferece as funcionalidades de *roaming* dentro de um ESS (*Extended Service Set*) e gerenciamento do consumo de energia. Dentro do gerenciamento de energia o IEEE 802 permite que as estações desliguem seus transceptores, economizando parte da energia consumida pelo equipamento. O protocolo da subcamada MAC é o CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*). [49]

4.2.5 Topologia

O padrão 802.11 suporta dois tipos de topologias: IBSS (*Independent Basic Service Set*) e ESS (*Extended Service Set*). Estas duas topologias utilizam como base o conceito de BSS (*Basic Service Set*), que define uma área onde as estações permanecem conectadas. Uma estação estará livre para se movimentar dentro do BSS, deixando de se comunicar com as outras estações quando deixar a área compreendida pelo BSS.

4.2.5.1 IBSS

Uma rede IBSS é composta de pelo menos dois computadores em um único BSS. Neste BSS não existe infraestrutura que ligue o BSS a outras rede.

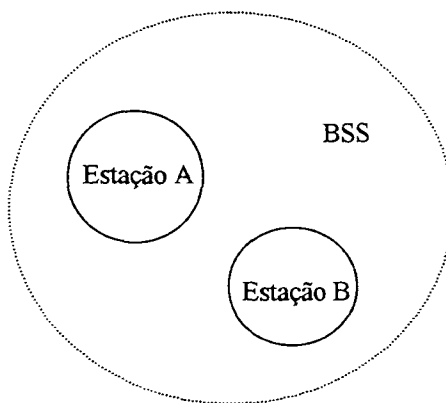


Figura 4.5 — IBSS o tipo mais básico de rede sem fio 802.11

4.2.5.2 ESS

Um ESS consiste em múltiplas células interconectadas através de pontos de acesso e uma infraestrutura de distribuição, como por exemplo *Ethernet*.

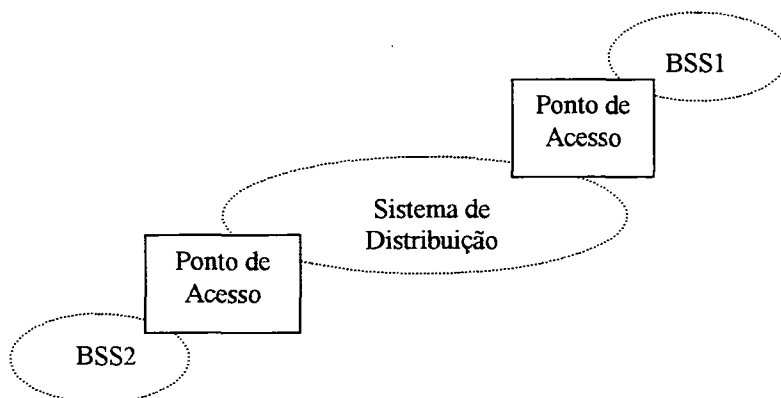


Figura 4.6 — ESS satisfaz a necessidade das redes de grande abrangência e maior complexidade

4.3 Arquitetura de Rede do Windows

Mesmo não seguindo fielmente o padrão OSI é possível fazer um paralelo entre as camadas propostas pelo modelo OSI e a arquitetura de rede da plataforma Windows, figura 4.2 [7].

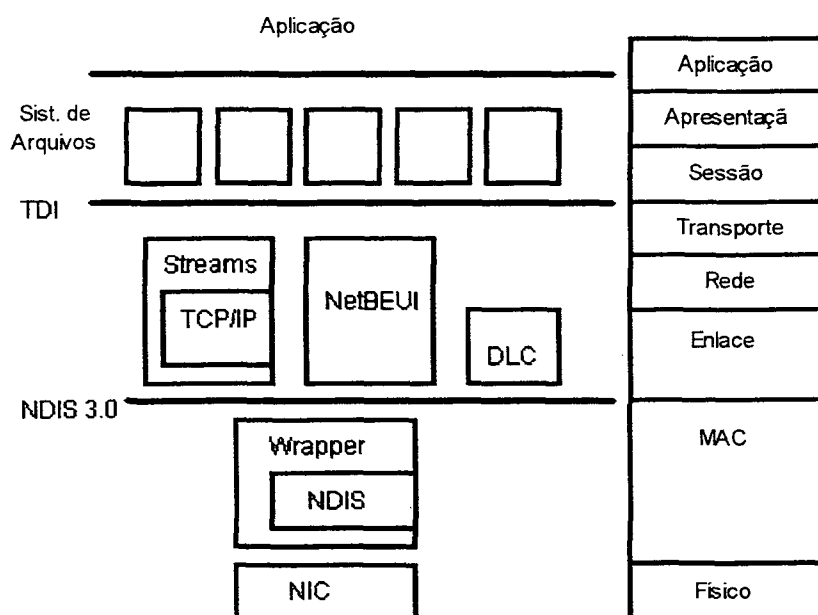


Figura 4.7 — Modelo OSI e a arquitetura de rede do Windows

Os drivers de rede da plataforma Windows implementam os protocolos e as interfaces das primeiras quatro camadas do modelo OSI.

A camada física é a camada mais baixa do modelo OSI. Esta camada é responsável pela recepção e pela transmissão de conjuntos de bits não estruturados através do meio físico. A camada física descreve as interfaces elétricas, óticas, mecânicas e funcionais para o meio físico. A camada física leva os sinais para todas as camadas mais altas do modelo OSI.

No Windows, a camada física é implementada pelo NIC (*Network Interface Card*), o seu transmissor/receptor e o meio ao qual está conectado. Para componentes de rede que utilizam a porta serial, a camada física pode também incluir o software responsável pela

definição de como o conjunto de bits é dividido em pacotes de dados antes de ser transferidos pela serial.

A camada de dados é dividida em duas sub-camadas: LLC e MAC. A sub-camada LLC provê transferências de *data-frames* de um ponto a outro livre de erros. Esta sub-camada é responsável por estabelecer e terminar links lógicos, controlando o *frame flow*, sequência de pacotes, reconhecimento de pacotes e retransmissão de pacotes com problemas. A LLC utiliza pacotes de reconhecimento e retransmissão para disponibilizar uma transmissão virtualmente livre de erros para as demais camadas que ficam acima.

A sub-camada MAC gerencia o acesso ao meio, verificação de erros nos pacotes, reconhecimento de endereços dos pacotes recebidos. Na arquitetura do Windows as funções reservadas a LLC são implementadas no driver de transporte, enquanto aquelas reservadas ao MAC são implementadas no NIC. O NIC é controlado por um driver de dispositivo chamado *NIC Driver*.

A camada de rede controla a operação da *subnet*. Ela decide qual caminho físico os dados deverão seguir baseado nas condições da rede, prioridade dos serviços e outros fatores. Isso inclui o roteamento, controle de tráfego, fragmentação e remontagem do pacote, mapeamento de endereços lógicos para endereços físicos e contabilização do uso da rede.

O objetivo da camada de **Transporte** é assegurar que as mensagens são entregues sem erros, na sequência correta e sem perdas ou duplicações. Ela retira dos protocolos das camadas mais altas a preocupação da organização na transferência dos dados entre as estações. A camada de transporte é responsável pela implementação de um protocolo

específico de rede como TCP/IP ou IPX/SPX, sendo independente do dispositivo de hardware utilizado pela rede.

No modelo de rede do Windows, a LLC, a camada de rede e a de transporte são implementadas por software em drivers chamados de *Transport Drivers*.

4.3.1 NDIS Drivers

Os drivers de rede no sistema operacional Windows suportam comunicação ponto a ponto e cliente servidor entre um nó local e um nó remoto em uma LAN ou WAN. Neste sistema operacional é utilizado drivers em um ou mais níveis da rede para passar pacotes de dados e fazer as necessárias traduções de protocolo, permitindo uma comunicação livre de erros entre os pontos de rede.

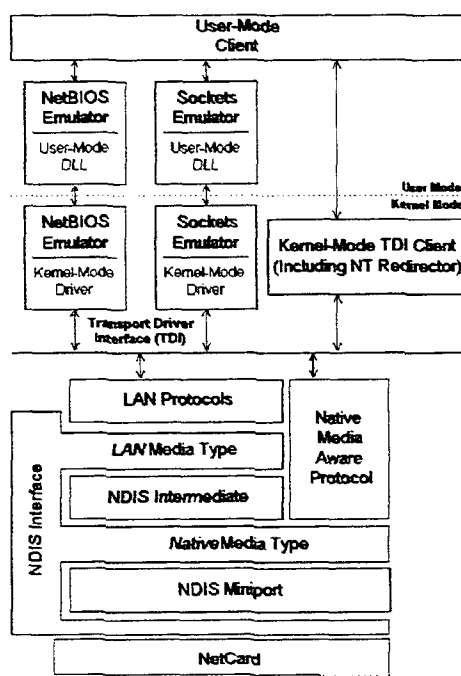


Figura 4.8 — Componentes existentes na arquitetura de rede do Windows

O Windows suporta três tipos de drivers de rede:

- *NIC Drivers (Network Interface Card);*
- *Intermediate protocol drivers;*
- *Upper level protocol drivers.*

4.3.1.1 NIC Drivers

NIC drivers gerenciam diretamente a placa de rede. Um NIC driver manipula diretamente o hardware e fornece uma interface para que as demais camadas da rede possam enviar pacotes de dados, manipular interrupções, inicializar ou parar a placa de rede e configurar as características do driver. Na plataforma Windows estes drivers podem ser do tipo *miniport* ou *legacy full NIC drivers* que é o modelo antigo de desenvolvimento de drivers de rede para Windows.

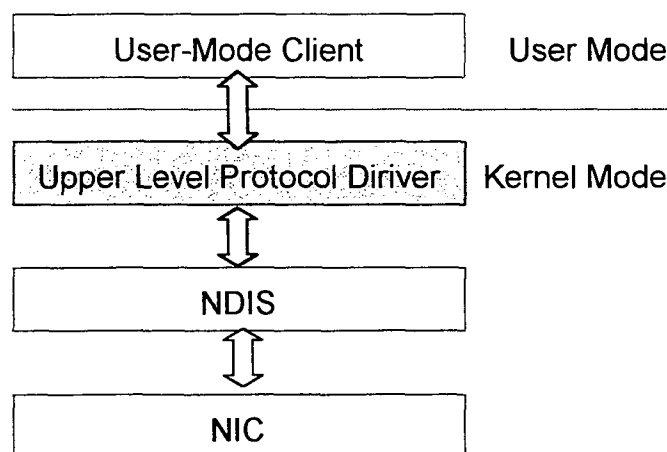
4.3.1.2 Intermediate protocol drivers

Um *Intermediate protocol driver* faz a interface entre um driver de nível mais alto, como um driver de transporte antigo, para um driver *miniport*. Para um driver de nível mais alto um *Intermediate driver* tem a mesma interface de um driver do tipo *miniport*. Para um driver do tipo *miniport* um *intermediate driver* exporta uma interface igual a um driver de protocolo. Um *intermediate driver* pode se ligar a outro *intermediate driver*.

Uma razão para o desenvolvimento de um *intermediate driver* seria para fazer a conversão entre um driver de transporte e um *miniport driver* que gerencia uma placa de rede para um novo meio de transmissão. Por exemplo, um *intermediate driver* poderia traduzir um protocolo de LAN para ATM.

4.3.1.3 Upper level protocol driver

Um *upper level protocol driver* implementa um interface entre a aplicação do usuário e as camadas de rede do Windows. O *upper level protocol* pode utilizar uma interface do tipo TDI, ou uma interface específica para trocar dados com as aplicações. Um driver deste tipo aloca memória para os pacotes recebidos das aplicações, copia estes dados da aplicação para o pacote alocado e envia este pacote para os drivers do nível mais baixo através de chamadas ao NDIS. Este driver também fornece uma interface para o recebimento de pacotes das camadas de nível mais baixo da rede, fazendo a distribuição destes pacotes para as suas respectivas aplicações do usuário.



4.4 Componentes "de prateleira"

Faz parte do objetivo do trabalho a utilização de tecnologias ou componentes de prateleira. O hardware utilizado no projeto pode ser enquadrado na categoria "prateleira" por ser vendido em lojas especializadas, normalmente voltadas a produtos para rádio-amadores.

A procura por uma solução que oferece de forma transparente a rede sem fio para as aplicações Windows, tem como resultado a possibilidade de utilizar qualquer um dos ambientes de desenvolvimento existentes neste sistema operacional. Ou seja, não é necessário a criação de nenhum novo ambiente de desenvolvimento ou uma nova biblioteca para a programação.

Redes sem fio sofrem com a ocorrência de desconexões, baixa velocidade de transmissão e segurança, problemas semelhantes aos encontrados em redes Internet. Por este motivo os navegadores para Internet possuem recursos como *time-out* para a carga de páginas, *cache* para a redução no tráfego de dados e comunicação usando criptografia. Conforme mostrado no item anterior, a rede sem fio será apresentada de forma transparente as aplicações no sistema operacional. Aproveitando esta liberdade de escolha do software, foi optado pelo navegador como ambiente de execução das aplicações a serem desenvolvidas para esta rede. A escolha do navegador não só permite que o desenvolvedor de sistemas possa escolher qualquer ambiente de desenvolvimento Internet, como também possa aproveitar os recursos de *cache* e segurança que estão embutidos no navegador e que são importantes para o funcionamento das aplicações que executam em redes sem fio.

4.5 Restrições

Para a montagem do protótipo estão disponíveis dois computadores com Windows instalado, dois rádios TEKK 2W 450 a 470 MHz e duas placas serial padrão Baycom (BAY96_V2).

Os recursos disponíveis restringem a implementação da rede a uma velocidade de 9600 bps. Para garantir o funcionamento do navegador e permitir a utilização de

ferramentas Internet o protocolo adotado é o TCP-IP. Foram apresentados neste capítulo algumas informações sobre o padrão IEEE 802.11 para redes sem fio. A implementação deste padrão não consiste apenas de regras para o desenvolvimento de camadas de software, o padrão também define as características necessárias ao hardware. O hardware a ser utilizado no projeto não suporta nenhuma das três camadas físicas apresentadas pelo IEEE 802.11 (*Frequency Hopping*, *Direct Sequence* ou *Infrared*). Por isso, a rede sem fio proposta não estará em conformidade com o padrão.

4.6 Arquitetura da Solução

Para a implementação de uma rede sem fio na arquitetura Windows transparente para as aplicações será necessário concentrar o trabalho na camada de enlace (LLC e MAC), o que na plataforma Windows significa desenvolver um driver do tipo *NDIS Miniport NIC driver*. Desenvolver o driver segundo a arquitetura de rede mostrado na figura 4.8/4.9 garante que as demais camadas do Windows possam utilizar os recursos disponibilizados pela rede sem fio. Isso fará com que o uso da rede sem fio pelo Windows seja simples, porque ela se apresentará ao usuário/sistema operacional como uma rede do tipo *Ethernet*. Isso garante que qualquer ferramenta de desenvolvimento Windows incluindo, portanto, as de Internet tenha acesso aos recursos da rede sem fio.

Agrupando as informações apresentadas nos itens anteriores deste capítulo, a solução proposta para o experimento de rede sem fio é apresentada na figura 4.9.

O sistema operacional será o Windows 95 e mesmo que seja possível utilizar o protocolo NetBEUI, TCP/IP e IPX/SPX o protocolo escolhido foi o TCP/IP.

A escolha de focar a implementação da rede sem fio no desenvolvimento de apenas um driver, tem o objetivo de baratear ao máximo o custo da solução, que depende de um hardware barato, composto de rádio e placa serial, e de um sistema operacional tradicional. O protocolo TCP/IP tem um comportamento razoável em ambientes de baixa velocidade, o que combina com a velocidade de 9600bps fornecido pelo equipamento utilizados neste projeto. Além disso, confirma a proposta de utilizar as ferramentas de desenvolvimento Internet na criação de sistemas. Isso também tem influência sobre o custo, porque não exige conhecimentos adicionais para quem implementar um sistema nesta rede.

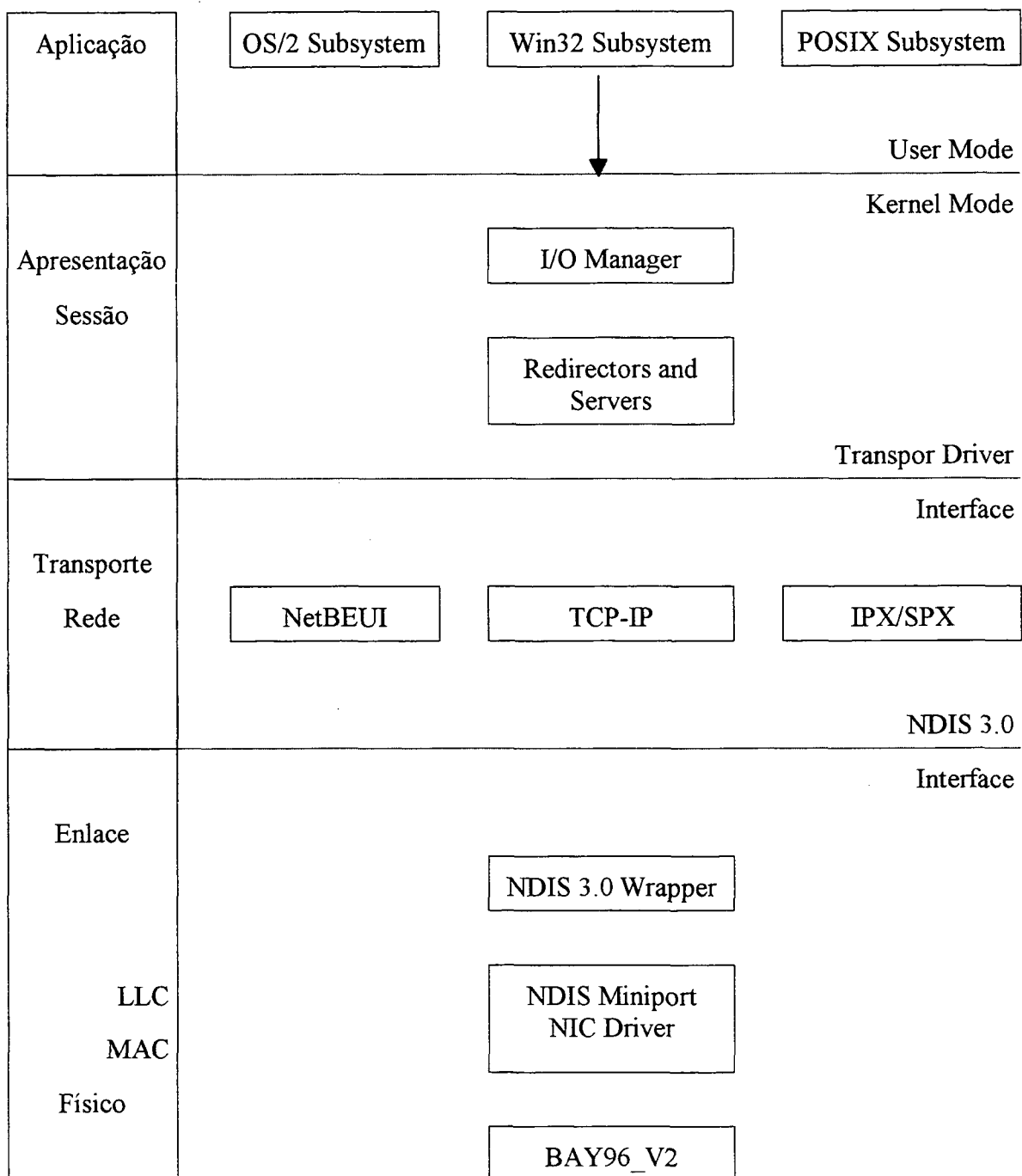


Figura 4.9 — Arquitetura geral da solução

4.7 Desenvolvendo o NDIS Miniport NIC driver

Desenvolver um driver do tipo *NDIS Miniport NIC driver* significa trabalhar com as camadas física (PHY) e de enlace (LLC/MAC) do IEEE 802. A figura 4.10 apresenta os módulos que precisam ser programados para a implementação do driver.

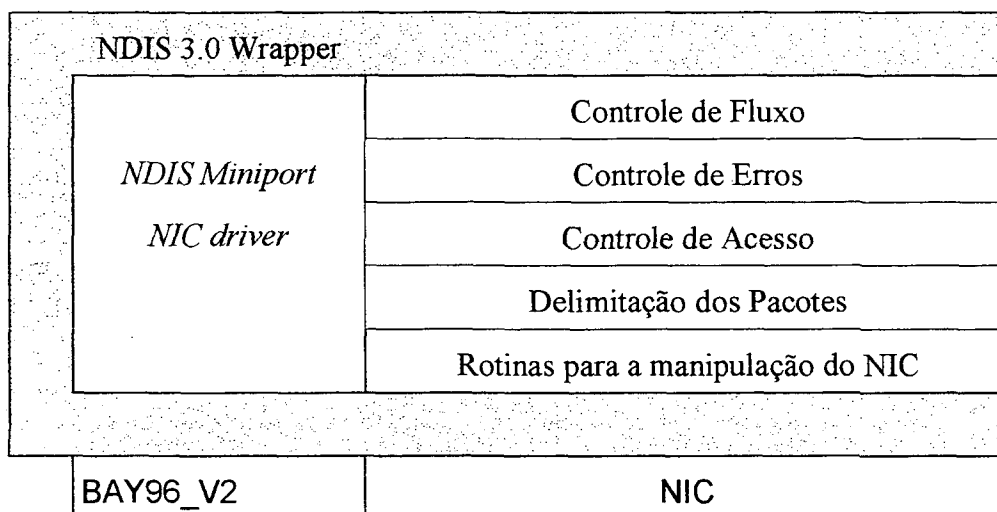


Figura 4.10 — Componentes a serem programados no NDIS Miniport NIC driver

4.7.1 Nível Físico - NIC

O nível físico é a primeira camada do modelo OSI, responsável pelo envio de um conjunto não estruturado de bits através de algum meio físico, neste caso ondas de rádio, assegurando através do seu protocolo físico que se a estação transmissora enviar, por exemplo, um bit 1 a estação receptora reconhecerá este bit 1. O nível físico fornece as características mecânicas, elétricas, funcionais e de procedimentos para ativar, manter e desativar conexões físicas para a transmissão de bits. Esta camada define como o cabo é ligado na interface de rede, como por exemplo a quantidade de pinos que o conector deve ter e qual a função de cada um destes pinos. Além disso, esta camada também define qual a técnica que será utilizada para o envio de dados, como *half-duplex* ou *full-duplex*.

Neste trabalho foi utilizado como NIC uma placa serial do tipo Baycom (BAY96_V2) da Microlaser Sistemas Optoeletrônicos [8]. As características desta placa podem ser encontradas no Apêndice A.

No experimento foi utilizado o quarto canal da placa (canal 03) que possui modem FSK podendo funcionar entre 2400 a 38400 bps. Como é utilizada apenas uma frequência de rádio para a transmissão e para a recepção de dados, a comunicação será do tipo *half-duplex*.

4.7.2 Rotinas para a manipulação do NIC

Todo o apêndice A está dedicado a explicação da programação da placa para o envio e recepção dos dados pelo rádio.

4.7.3 Delimitação dos Pacotes

Para a implementação do driver é necessário colocar algum mecanismo que permita o reconhecimento das informações transmitidas. A partir deste mecanismo será possível determinar onde, dentro de um conjunto de bytes recebidos, se encontram os pacotes de informação. Este controle será feito por software, utilizando como informação a portadora e o byte SFD (*Start Frame Delimiter*). Sempre que o placa receber o sinal da portadora (RTS) o driver iniciará a sua procura pelo início do pacote, ou seja o byte SFD. Mais detalhes no capítulo 5, item Enviando e Recebendo Pacotes.

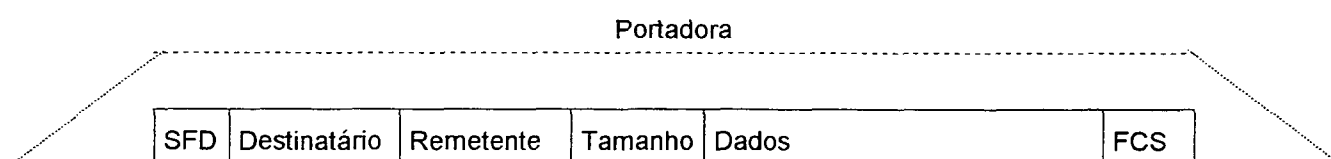


Figura 4.11 — Identificação do pacote

4.7.4 Nível de Enlace – Controle de Acesso

Como foi mostrado nos itens anteriores, o padrão IEEE 802 dividiu a camada de enlace em duas subcamadas (LLC e MAC). A manipulação das características específicas das várias tecnologias ficou sob a responsabilidade da subcamada MAC, que trata do controle de acesso ao meio. As demais atividades da camada de enlace ficaram a cargo da subcamada LLC.

O gerenciamento de acesso ao meio consiste em controlar a comunicação de dados entre as estações que estão compartilhando o mesmo meio de acesso. Como no experimento prático está sendo utilizando para a transmissão e recepção de dados um rádio operando em apenas uma frequência, significa que todas as estações que estiverem conectadas a esta rede terão que compartilhar o mesmo meio de transmissão. Por isso alguma técnica precisa ser utilizada para disciplinar as transmissões de forma a evitar a ocorrência de colisões, que acontecerão quando mais de uma estação estiver enviando dados ao mesmo tempo.

Existem diversas estratégias, como por exemplo CSMA, *Aloha*, *Polling* e *Slot*. [2], utilizadas para controlar o acesso ao meio. A estratégia escolhida para o controle de acesso a nossa rede sem fio é a mesma definida pelo padrão IEEE 802.11, ou seja, CSMA/CA [4]. O DFWMAC (*Distributed Foundation Wireless Media Access Control*) apresentado pelo IEEE suporta dois métodos de acesso: distribuído e centralizado:

- **Distribuído**, as estações é que decidem quando transmitir, podendo ocorrer colisões.
- **Centralizado**, a decisão é centralizada em um ponto que determina qual a estação deve transmitir, evitando as colisões.

A método escolhido foi o distribuído para que não seja necessário a existência de uma estação responsável pela organização do processo de transmissão dentro do BSS.

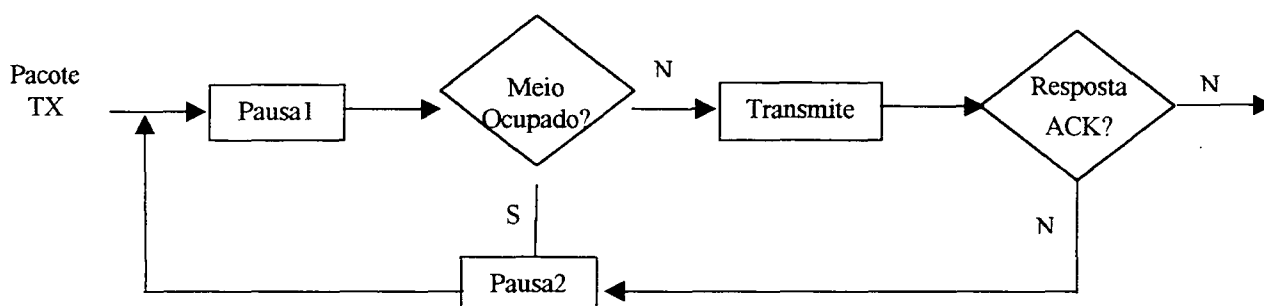


Figura 4.12 — Resumo do funcionamento do CSMA/CA

Como mostrado na figura 4.12 o princípio básico do funcionamento do CSMA/CA é o de tentar evitar a ocorrência das colisões durante a transmissão. Neste método de acesso, quando deseja transmitir, a estação "ouve" antes o meio para saber se existe alguma transmissão em progresso. Esta escuta pode ser feita através da verificação da existência de uma portadora. Se nesta escuta ninguém estiver utilizando o meio, a estação pode transmitir, caso contrário a estação espera por um período de tempo (Pausa2) antes de tentar novamente. Outro motivo que pode fazer com que a Pausa2 seja necessária é a falta do recebimento de um *ack* confirmando a correta recepção da mensagem pela estação destino.

O valor do tempo Pausa1 e Pausa 2 é apresentado a seguir:

- **Pausa1.** A maior parte dos pacotes transmitidos possuem um campo especial chamado **duração** quem tem o objetivo de identificar o tempo de duração do

processo de transmissão dos dados. Ao receber um pacote transmitido por outra estação, o campo **duração** é guardado servindo como um relógio para indicar quando a transmissão atual estará encerrada. Esta pausa evita que as estações tentem transmitir enquanto duas estações, que já tem o direito de acesso ao meio, estejam trocando dados.

- **Pausa2.** O objetivo do tempo **pausa2** é o de diminuir a probabilidade da ocorrência de colisões. O momento que sucede a liberação do meio para outras transmissões possibilita que duas ou mais estações verifiquem a portadora ao mesmo tempo, consequentemente fazendo com que a tentativa de reservar o canal para transmissão também aconteça ao mesmo tempo, gerando as colisões. A pausa consistirá na geração de valores calculados a partir do número de tentativas de transmissão e no seu intervalo de tempo pré-alocado. Para garantir que a semente utilizada no sorteio seja diferente para cada estação, é possível utilizar o número de identificação da estação como valor inicial.

4.7.5 Controle de Erros/Fluxo

A escolha do protocolo TCP (*Transmission Control Protocol*) proporciona um serviço confiável de transferência de dados. O TCP foi projetado para funcionar como base de um serviço de rede que não utiliza conexão e que não possui confirmação. O TCP não exige um serviço de rede confiável para funcionar, responsabilizando-se pela recuperação dos dados corrompidos, perdidos, duplicados ou entregues fora de ordem. Mesmo assim estão sendo embutidos no driver controles para garantir uma transmissão mais confiável, através do uso de CRC's e pacotes especiais para controle das transmissões.

4.7.5.1 CRC

O controle de erros está sendo implementado a partir do cálculo de um CRC (*Cyclic Redundancy Check*) sobre o bloco de informações que estão sendo transmitidas. O algoritmo utilizado gera um valor de 32 bits baseado no *AUTODIN II CRC algorithm* utilizados nas placas NE1000/NE2000 e que faz parte do padrão *Ethernet*.

4.7.5.2 Pacotes especiais

Com o objetivo de organizar o fluxo de informações entre as estações, a figura 4.13 apresenta como foi utilizado alguns pacotes especiais na transmissão, chamados pacotes de controle. São eles:

- **RTS** (*Request to Send*), sempre que uma estação desejar enviar algum dado a outra estação, primeiro a estação origem deverá solicitar uma autorização através do envio de uma mensagem de controle RTS para a estação destino. A estação origem somente poderá iniciar a transmissão depois que receber uma autorização da estação destino.
- **CTS** (*Clear to Send*), este pacote de controle é enviado pela estação destino para indicar que uma transmissão solicitada pode ser iniciada.
- **ACK**, este pacote é enviado pela estação destino para indicar a recepção de algum conjunto de dados. Se a estação de origem não receber esta mensagem, ela transmite o pacote de dados novamente, considerando que ocorreram problemas na recepção do pacote pela estação destino.

Este tipo de controle de fluxo possui uma exceção para as mensagens do tipo *broadcast* e *multicast*. Como este tipo de mensagem possui diversos destinatários, existe

um problema no envio da confirmação da transmissão. Dentro da topologia de rede apresentada a seguir, não existe a figura de uma estação principal ou servidora, por isso o problema foi resolvido usando uma estratégia simples. Para as mensagens do tipo *broadcast* e *multicast* não é utilizado o esquema de mensagens de controle RTS/CTS/ACK, dando o direito da estação transmitir o pacote diretamente. Existem estudos sobre a otimização do envio de mensagem em redes sem fio, alguns deles podem ser encontrados em [20], [19], [18], [19] e [14].

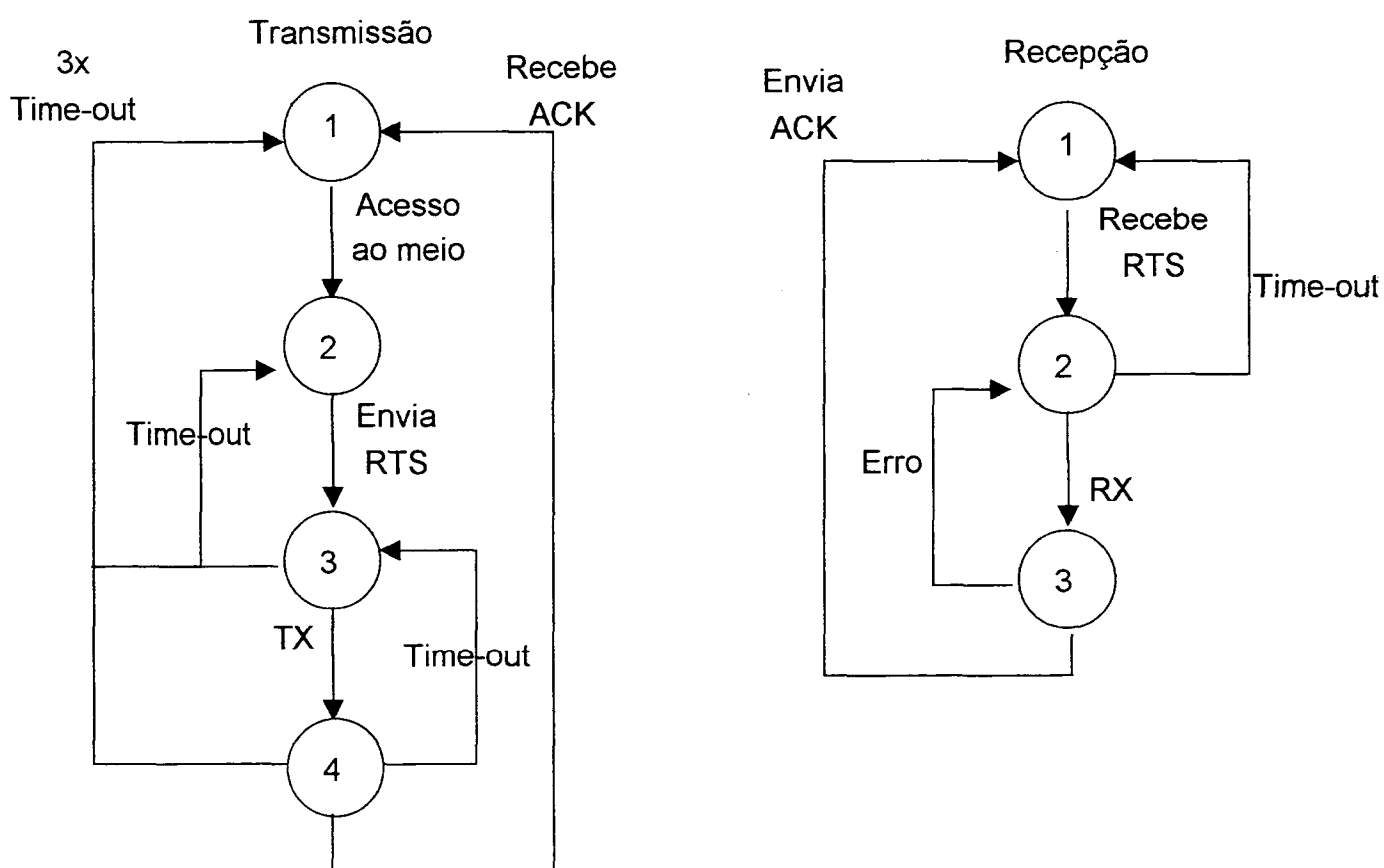


Figura 4.13 — Diagrama das operações de Transmissão e Recepção

4.7.5.3 Pacotes de dados

Uma das informações que o sistema operacional solicita ao driver está relacionada ao tipo da rede. O driver proposto nesta dissertação é do tipo *Ethernet*. Trabalhar com este tipo de rede significa manipular pacotes com tamanho de 1500 bytes. Considerando uma velocidade de transmissão máxima de 9600 bps e um meio sujeito a diversas fontes de interferência que ocasionam a inserção de ruídos na transmissão, o tempo para a retransmissão de um pacote de 1500 bytes é significativo na performance geral da rede. Através da utilização da função `QueryInformation` [7] pode-se reconfigurar o tamanho do pacote para 256 bytes, aumentando o tempo para a transmissão das mensagens mas reduzindo o tempo das retransmissões. O tamanho de 256 foi o tamanho máximo do pacote escolhido para a implementação do driver. O número 256 não foi escolhido ao acaso, está baseado no tamanho padrão de pacote utilizado pelo protocolo AX25, muito usado em ambientes de transmissão com rádio [50]. Neste trabalho não foi feito um estudo para avaliar a performance da rede e a sua relação com o tamanho do pacote utilizado.

Não foi feito nenhum estudo ou experimento para verificar qual tamanho de pacote seria mais adequado para esta rede. Para um melhor aproveitamento da transmissão na rede o ideal é a utilização de um tamanho de pacote variável, que se ajuste conforme a qualidade do meio, ou seja, o número de retransmissões que estão sendo necessárias. Este tipo de característica não está sendo implementado neste driver de rede.

Outra forma de garantir uma melhor performance na comunicação utilizando a rede sem fio é o desenvolvimento de outro driver de transporte que seja mais adequado as características da comunicação sem fio do que o TCP/IP. O AX25, ou a proposta do driver RPWNT [16], são alternativas interessantes.

4.7.6 Topologia

Conforme o padrão IEEE 802.11 apresentado neste capítulo existem duas topologias disponíveis: IBSS (*Independent Basic Service Set*) e ESS (*Extended Service Set*). A topologia escolhida para a rede foi a IBSS. Mesmo assim, através de configurações no sistema operacional é possível transformar a rede do tipo IBSS para ESS, com a exceção do recurso de mobilidade entre os BSS existentes. A configuração consiste em ligar uma das estações do BSS a rede sem fio e a um sistema de distribuição usando cabos. A partir do roteamento dos pacotes do sistema de distribuição para a rede sem fio, e vice versa, esta estação passará a fazer o papel de um ponto de acesso, ligando as demais estações deste BSS a outros BSS ligados ao mesmo sistema de distribuição (figura 4.6).

De acordo com o DFWMAC (*Distributed Foundation Wireless Media Access Control*) apresentado pelo IEEE o método escolhido foi o distribuído para que não seja necessário a existência de uma estação responsável pela organização do meio dentro do BSS.

5. IMPLEMENTAÇÃO

O desenvolvimento deste trabalho sobre a implementação de uma rede sem fio teve duas etapas distintas. Na primeira foi feito um estudo com o objetivo de encontrar um caminho viável para o desenvolvimento desta rede. A segunda etapa, registrada neste capítulo, colocou em prática o estudo e procurou validar a estratégia de desenvolvimento apresentada no capítulo 4 através de um protótipo. Neste protótipo foram utilizados os seguintes recursos:

- 1 computador Pentium II 300 com Windows 98;
- 1 computador Pentium 133 com Windows 95;
- 1 computador Pentium III 233 com Windows 98;
- 2 dois rádios TEKK 2W 450 a 470 MHz;
- 2 placas serial padrão Baycom (BAY96_V2);
- 2 placas de rede NE2000;
- Microsoft Visual Studio;
- Microsoft SDK e DDK.

Os dois primeiros computadores foram utilizados para o teste do funcionamento da comunicação sem fio. Um computador com o Windows 98 e outro com o Windows 95 de forma a verificar o funcionamento do driver em duas versões diferentes do sistema operacional. O terceiro computador foi utilizado para o desenvolvimento do driver e para realização das operações de *debug*.

A implementação da rede foi um processo que incluiu muita "tentativa e erro". O capítulo 4 apresentou o caminho para o desenvolvimento (*NDIS Miniport NIC driver*) todavia o "como" fazer era algo desconhecido. Por este motivo ao invés de procurar resolver o problema diretamente, ou seja, desenvolver o *NDIS Miniport NIC driver*, o trabalho foi organizado em uma série de etapas procurando aliar a aprendizagem dos componentes usados no trabalho (placa Baycom e o desenvolvimento de drivers do tipo *kernel*) com a implementação da rede sem fio (programação do *NDIS Miniport NIC driver*).

Estas foram as etapas:

- **Como a placa funciona?** Desenvolvimento de um programa que inicializa a placa, finaliza a placa e faz as operações de envio e recebimento de dados no modo *loopback* [24].
- **Comunicando através de cabos.** Desenvolvimento de um programa que faz a comunicação de dados entre dois computadores usando a placa Baycom ligada por meio de cabos.
- **Entendendo o funcionamento de um *NDIS Miniport NIC driver*.** Recompilação do código fonte do driver NE2000, e a realização de testes para verificar o seu funcionamento.
- **Enviando e recebendo pacotes.** Desenvolvimento de um programa que faz o envio de pacotes entre dois computadores ligados por cabos utilizando a placa Baycom.
- **Primeira versão do *NDIS Miniport NIC driver*.** Desenvolvimento do driver de rede que faz a comunicação usando a placa Baycom ligada através de cabos.

- **Enviando e recebendo pacotes através do rádio.** Desenvolvimento de um programa que faz o envio de pacotes entre dois computadores usando a comunicação por rádio.
- **Segunda versão do *NDIS Miniport NIC driver*.** Desenvolvimento de uma nova versão do driver de rede, agora com a comunicação via rádio.
- **Terceira versão do *NDIS Miniport NIC driver*.** Desenvolvimento de uma nova versão do driver. Nesta versão foi feita uma pequena otimização no protocolo utilizado pelo driver.

5.1 Como a placa funciona?

A programação da placa Baycom exige a manipulação de interrupções e endereços de memória. Estas operações são possíveis em programas Windows, mas em nome da segurança e organização dos programas que estão sendo executados no computador, o sistema operacional possui uma série de procedimentos que precisam ser cumpridos pelos programas, principalmente aqueles que fazem a manipulação de endereços e interrupções.

A meta desta etapa foi entender o funcionamento da placa. Para não fugir deste propósito, ao invés de fazer um programa para Windows, foi escolhido fazer um programa para DOS. Outra simplificação foi a programação da placa utilizando o modo *loopback*. Neste modo é possível testar os recursos de comunicação da placa em apenas um computador. Tudo o que é transmitido no modo *loopback* é recebido pela própria placa. O resultado deste estudo foi registrado no apêndice A.

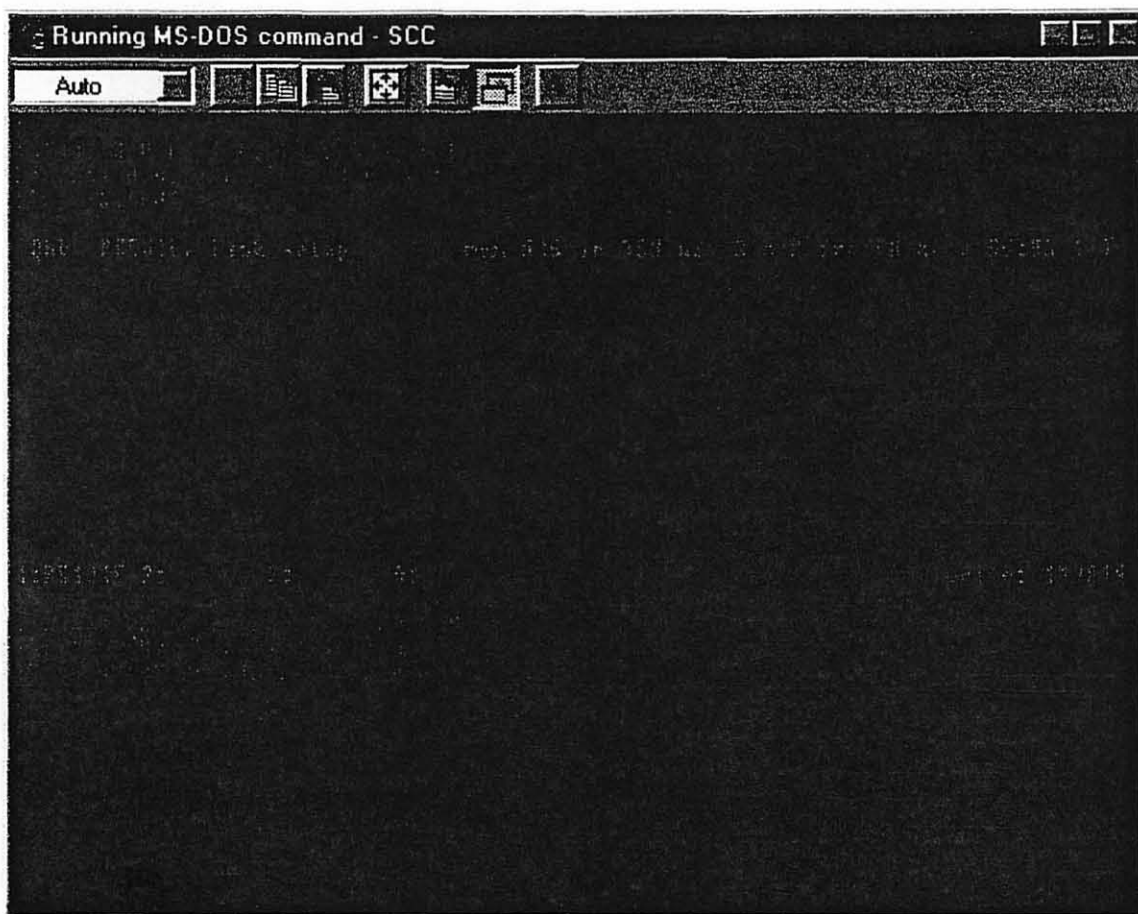
5.3 Comunicando através de cabos

No primeiro programa o trabalho foi direcionado para entender funcionamento da placa. A transmissão de dados não acontecia realmente, porque através do recurso de *loopback* todas as transmissões eram recebidas pela própria placa que transmitiu. Neste segundo programa (figura 5.2) é feito efetivamente a comunicação entre dois computadores. Para a realização deste experimento foi utilizado um cabo ligando diretamente as duas placas, permitindo uma comunicação *full-duplex*.

Ligação direta entre as duas placas:

USCC #1		USCC #2	
Pino	Descrição	Pino	Descrição
23	TX	24	RX
24	RX	23	TX
25	GND	25	GND

Para verificar o funcionamento dos cabos foi usado o programa **Baycom**, conhecido pelos rádio-amadores, que permite fazer a comunicação entre computadores utilizando o protocolo AX25 (figura 5.1).



**Figura 5.1 — Programa Baycom conectando com outro computador (PU1JIF)
utilizando cabos**

Este é o modo mais simples de funcionamento da placa, como a saída TX da primeira placa está ligada diretamente a entrada RX da segunda placa, não é necessário se preocupar com a portadora, basta escrever no endereço de dados da placa para fazer a transmissão. A existência de dois canais independentes de comunicação (*full-duplex*) permitiu que a implementação da comunicação fosse feita sem a utilização de nenhuma técnica para controlar o acesso ao meio [2].

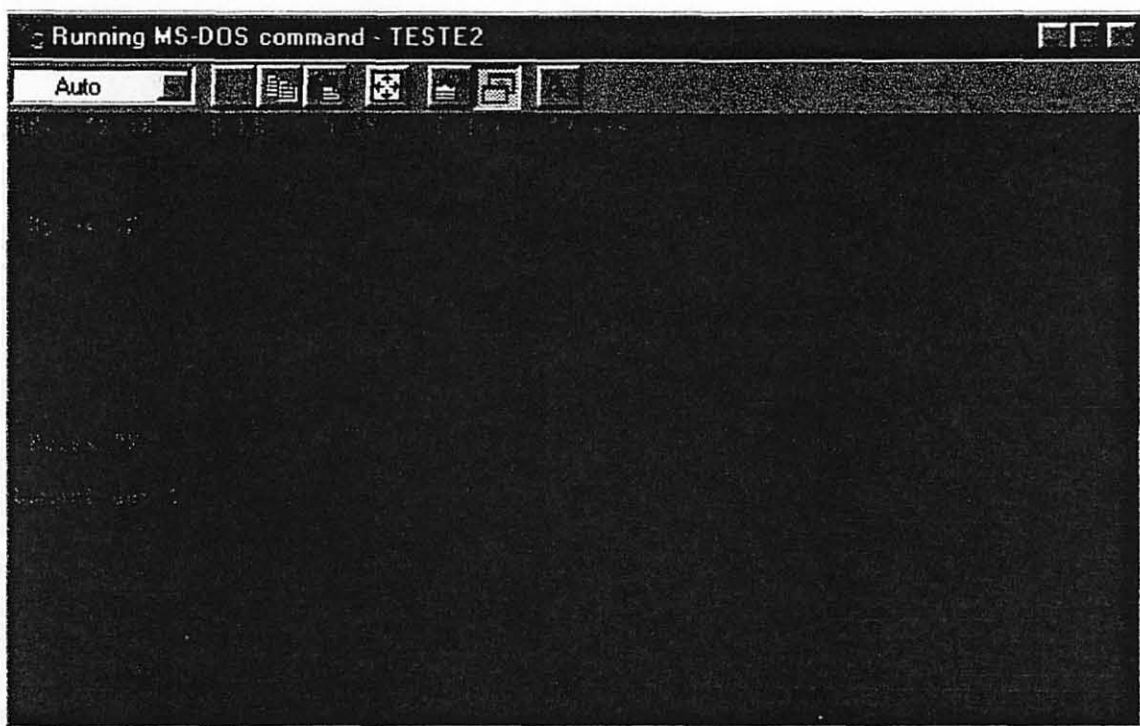


Figura 5.2 — Programa DOS que troca bytes entre dois computadores conectados com a placa ligada através de um cabo

5.4 Entendendo o funcionamento de um NDIS Miniport NIC driver

5.4.1 NDIS Miniport NIC driver

O desenvolvimento de drivers exige que o programador respeite uma série de regras do sistema operacional. Diferente de uma aplicação que vai executar no modo usuário, um driver é um componente que desfruta de privilégios, tendo o direito de executar instruções como as de acesso aos dispositivos ou as que permitem influenciar o funcionamento da memória virtual. Em contrapartida os componentes e a organização do sistema operacional são muito mais visíveis ao programador, exigindo um conhecimento mais profundo do seu funcionamento. A figura 4.9 mostra um pouco desta organização. Existem procedimentos especiais para a inicialização do driver, finalização do driver e alocação de memória. Não é

recomendado a utilização das funções da biblioteca padrão da linguagem C, por isso existe um conjunto grande de rotinas do sistema operacional para uso exclusivo dos programas que são executados no modo *kernel*, como os drivers. Existem rotinas para a alocação de memória, manipulação de *strings*, acesso ao registro do sistema, data e hora, etc. Existem configurações e recomendações especiais para o processo de compilação/link. Até mesmo o *debug* do driver possui características próprias.

Antes de iniciar a programação de um driver é necessário realizar uma extensa leitura sobre a arquitetura do sistema operacional e sobre o modo de programação dos drivers. A documentação da Microsoft [7] traz um boa parte destas informações, entretanto referências como [1] e [13] são de leitura obrigatória para o aprendizado deste modo de programação.

Contudo, estes conceitos tem apenas caráter introdutório. Além do básico sobre desenvolvimento de drivers, a programação de drivers de rede exige o conhecimento da especificação NDIS (*Network Driver Interface Specification*, veja figuras 4.8 e 4.9). A especificação NDIS pode ser encontrada na literatura sobre o desenvolvimento de drivers de rede, entretanto a implementação de drivers de rede especificamente para a plataforma Windows é um assunto bastante restrito. A única fonte de informações documentadas sobre o desenvolvimento de *NDIS Miniport NIC drivers* que foi encontrada estava no Microsoft DDK (*Device Driver Kit*) [7].

Um roteiro para quem deseja aprender sobre o desenvolvimento de drivers de rede consiste, como citado, na leitura de pelo menos uma das referências [1] e [13]. Este estudo servirá como introdução a arquitetura do sistema operacional Windows e sobre o

desenvolvimento de drivers. O segundo passo é a leitura da referência [3], este livro apresenta uma visão geral sobre o desenvolvimento de drivers de rede. O livro descreve o NDIS, mas não mostra detalhes sobre como ele deve ser programado. Por este motivo o último passo consiste em estudar a documentação original da Microsoft sobre o desenvolvimento de drivers de rede [7].

5.4.2 Código fonte NE2000

O número de detalhes envolvidos na programação de um driver é muito grande. A documentação sobre o assunto apresenta o roteiro básico, mas não entra nos detalhes da implementação. É praticamente impossível desenvolver um driver de rede sem ter como base um programa de exemplo. O DDK da Microsoft vem com o código fonte de alguns drivers de rede, como por exemplo redes do tipo *Ethernet* ou *Token Ring*. A estrutura da rede sem fio proposta neste trabalho se assemelha ao funcionamento das redes do tipo *Ethernet* onde, entre outras características, quando ligadas em um barramento, a transmissão de uma estação é recebida por todas as estações ligadas a este barramento. Por este motivo, e também pela facilidade de se encontrar placas NE2000, o programa base para o estudo do desenvolvimento do *NDIS Miniport NIC Driver* foi o código fonte do driver de rede para placas NE2000.

Este experimento consistiu em preparar o ambiente de desenvolvimento [24] [1] [13] e recompilar o código fonte do driver para a placa NE2000. Para testar o funcionamento do driver foi preparado dois computadores (Win95 e Win98) com uma placa de rede NE2000. Após a instalação do driver original da placa, existente no sistema operacional, substituímos o arquivo NE2000.SYS presente no diretório

\WINDOWS\SYSTEM pelo novo arquivo NE2000.SYS gerado. Depois da reinicialização do computador foi usado o ambiente de rede do Windows para fazer cópias de arquivos entre os dois computadores.

Um estudo mais aprofundado do driver foi feito a partir do uso do software **SoftIce** (veja apêndice C). Outra estratégia usada para o estudo do código fonte NE2000 foi a criação de funções que, através da serial, imprimem mensagens descrevendo o funcionamento das rotinas internas do driver [12]. Para o acompanhamento destas mensagens foi conectado um terceiro computador na porta serial de um dos micros utilizados no teste. Neste terceiro computador estava presente o programa **Terminal** (Windows 3.1) monitorando a porta serial para apresentar as mensagens enviadas pelo driver.

5.5 Enviando e Recebendo Pacotes

Nos testes anteriores efetuados com a placa, a comunicação utilizada foi byte a byte. O protótipo de rede sem fio que está sendo montado enviará pacotes seguindo o formato *Ethernet*. Neste experimento o programa apresentado no item 5.3 foi modificado de forma a enviar pacotes de dados de acordo com o formato *Ethernet* (apêndice B). A diferença nesta implementação foi a modificação dos bits de preâmbulo e dos bits delimitadores de início de pacote. No lugar destes 8 bytes utilizamos apenas um byte com o objetivo de indicar o início do pacote que está sendo transmitido (SFD - *Start Frame Delimiter*).

Preâmbulo	SFD	Destinatário	Remetente	Tamanho	Dados + PAD	FCS
-----------	-----	--------------	-----------	---------	-------------	-----

Figura 5.3 — Pacote Ethernet

SFD	Destinatário	Remetente	Tamanho	Dados	FCS
-----	--------------	-----------	---------	-------	-----

Figura 5.4 — Pacote usado no experimento

Outra mudança foi a retirada do PAD, incluído no pacote *Ethernet* para garantir um tamanho mínimo na mensagem. Isso não é necessário nesta experimento utilizando cabos. A comunicação é *full-duplex* e não existe compartilhamento do meio para a transmissão ou recepção, não existindo a possibilidade de colisões. O cálculo para o FCS (*Frame Check Sequence*) foi o mesmo encontrado no código fonte do driver NE2000, ou seja, um valor de 32 bits baseado no *AUTODIN II CRC algorithm*.

5.6 Primeira versão do NDIS Miniport NIC driver

Baseado no resultado obtido pelos dois experimentos (5.4 e 5.5) foi construído uma primeira versão do driver de rede utilizando as placas ligadas por cabo. O objetivo deste programa foi verificar o funcionamento do driver de rede Windows utilizando os recursos da placa Baycom. Para simplificar o processo de *debug* o código foi organizado em duas camadas distintas: uma geral, contendo um código fonte compatível com o DOS e com o Windows; outra camada específica, uma parte exclusiva ao programa de teste DOS e outra contendo o código do driver Windows.

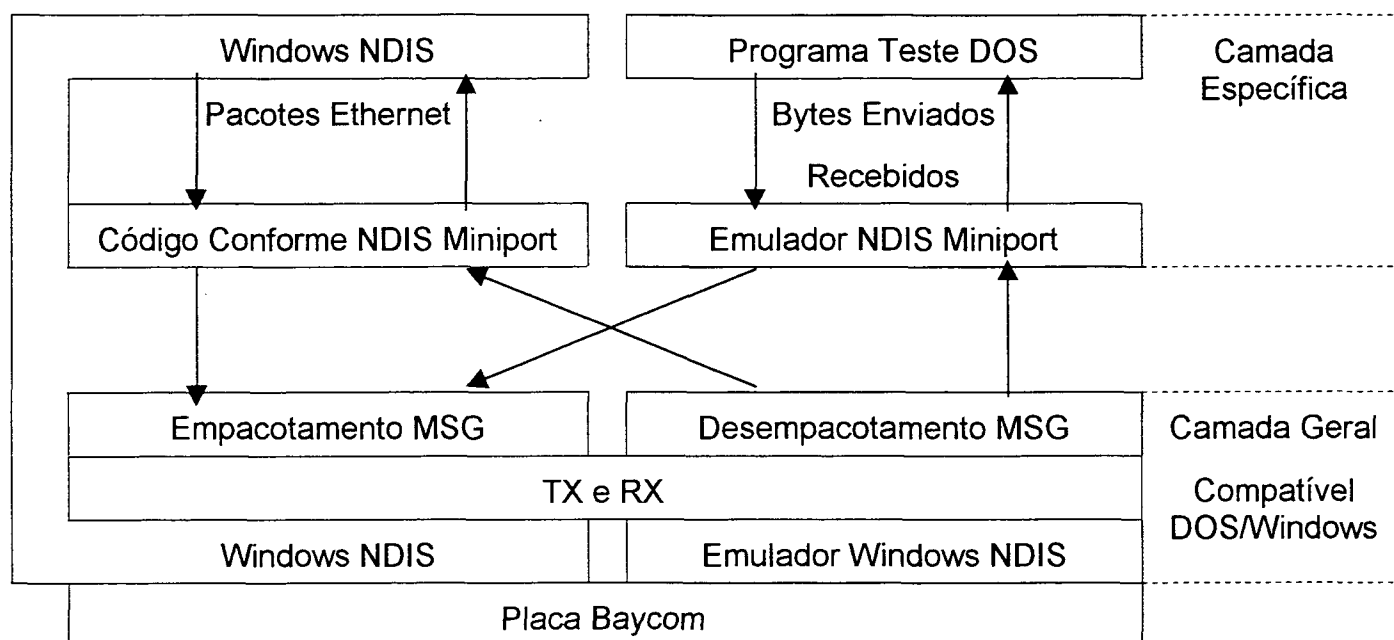


Figura 5.5 — Organização da primeira versão do NDIS Miniport

A partir desta organização foi possível montar parte do código do driver através de um programa DOS, simplificando o processo de *debug* destas rotinas.

Para testar o funcionamento do driver foram preparados dois computadores (Win95 e Win98) com a placa Baycom. Como ainda não tinha sido feito um *script* para automatizar a instalação deste novo driver (veja apêndice D), foi utilizado o processo de instalação da placa NE2000. Após a instalação do driver original da placa NE2000 substituímos o arquivo NE2000.SYS presente no diretório \WINDOWS\SYSTEM pelo novo driver gerado neste experimento. Depois da reinicialização do computador foi usado o ambiente de rede Windows para fazer cópias de arquivos entre os dois computadores. No teste foi feita a cópia de um arquivo de 1Mbyte, o tempo para a realização deste cópia foi de aproximadamente cinco minutos. Isso indica que mesmo que a placa tenha sido inicializada para 9600, a velocidade atingida pela comunicação ficou em torno de 4800 bps. O tamanho

máximo dos pacotes transmitidos seguiu o padrão *Ethernet* de 1514 bytes. Ao final da cópia os arquivos foram comparados com o programa FC.EXE existente no Windows.

5.7 Enviando e recebendo pacotes através do rádio

A última implementação que precisa ser feita para viabilizar a construção do driver para a rede sem fio, consiste na substituição da comunicação usando cabo pela comunicação usando o rádio. Para mudar da placa para o rádio duas modificações precisam ser feitas, a primeira consiste em alterar a forma de transmissão dos pacotes de *full-duplex* para *half-duplex*. A segunda é passar a controlar a portadora durante a transmissão. Antes de transmitir é necessário "levantar" a portadora (sinalização RTS/DCD) e ao final da transmissão "baixar" a portadora.

Cabo para o rádio:

USCC		Rádio	
Pino	Descrição	Pino	Descrição
22	PTT	03	PTT
23	TX	04	Data in
24	RX	05	Data out
25	GND	2	GND
		1	12V DC

Para a verificação do correto funcionamento dos cabos também foi utilizado o programa **Baycom**. Além disso também foi modificado o programa do item 5.5 para incluir o controle *half-duplex* na transmissão.

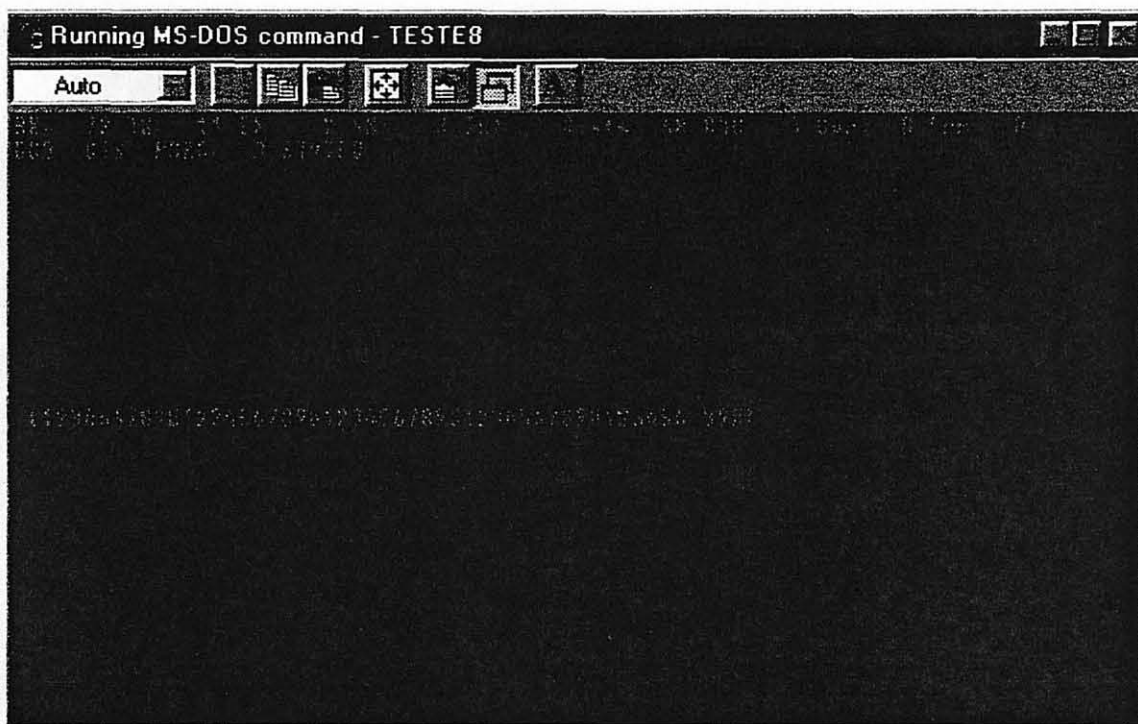


Figura 5.6 — Programa DOS que troca pacotes entre dois computadores ligados por rádio

5.8 Segunda versão do NDIS Miniport NIC driver

O método de acesso de uma rede *Ethernet* está baseado no CSMA/CD [2]. Conforme apresentado no capítulo 4 o controle de acesso ao meio do IEEE 802.11 é o DFWMAC (*Distributed Foundation Wireless Media Access Control*). Neste experimento foi alterado a primeira versão do driver acrescentando mais um módulo responsável pela implementação do controle DFWMAC.

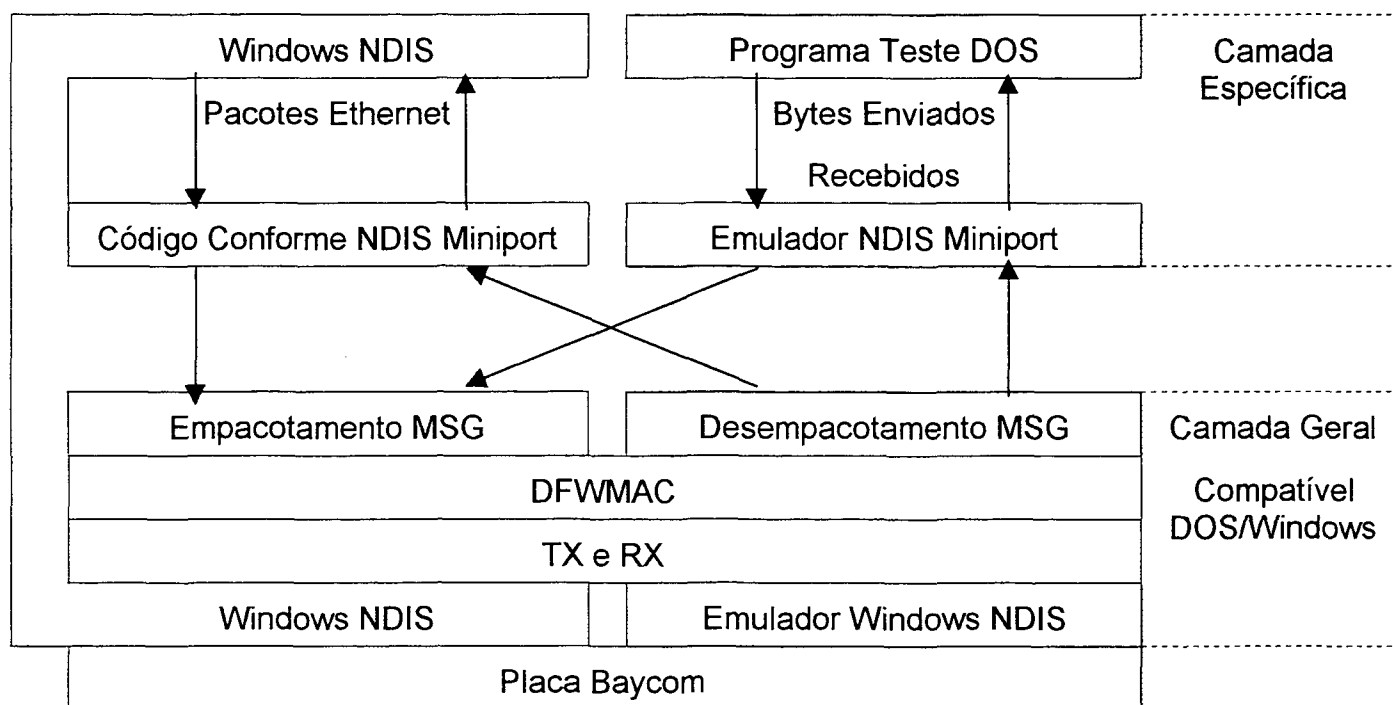


Figura 5.7 — Organização da segunda versão do NDIS Miniport

Para a organização da transmissão *half-duplex* o driver passou a utilizar os pacotes de controle RTS, CTS e ACK.

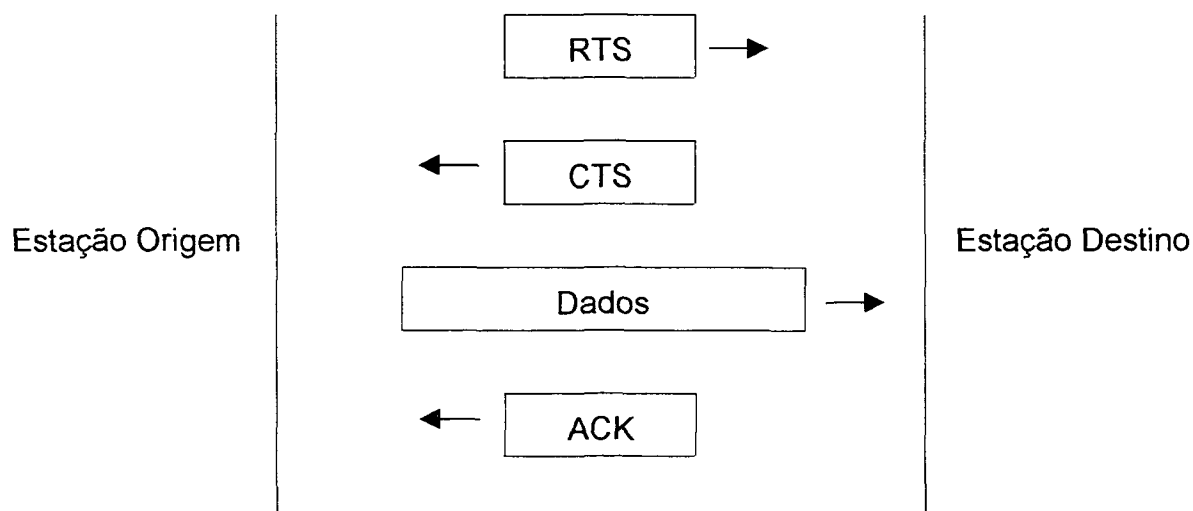


Figura 5.8 — Sequência de pacotes para a comunicação de dados

5.8.1 Teste através do ambiente de rede

Para testar o funcionamento desta nova versão do driver foi escolhida a mesma técnica utilizada no experimento descrito no item 5.6, a cópia de arquivos através do ambiente de rede. Antes de fazer a cópia foi utilizado o comando "Localizar → Computador" do Windows para procurar o computador através da rede sem fio. Neste experimento o resultado foi diferente do obtido pelo driver que utilizava a comunicação por cabos. O Windows não encontrou o computador na rede. Analisando a transmissão dos pacotes através do programa **NDIScope** foi possível perceber que ocorreu um *time-out* durante a operação de troca de pacotes entre os computadores. O motivo foi o aumento significativo no tempo de envio dos pacotes pela mudança de *full-duplex* para *half-duplex*. Para tentar resolver este problema de *time-out* foi feita uma pequena modificação no driver. Na rotina (`QueryInformation` [7]) que especifica as características do driver de rede o valor que configura a velocidade de transmissão da rede foi reduzido ao seu valor mínimo, ou seja 100 bytes por segundo. Mesmo assim o problema do *time-out* não foi resolvido porque os pacotes especiais enviados pela rede Windows para o reconhecimento dos computadores na rede (operação chamada *browsing*) eram menores que 200 bytes. O processo de uma transmissão (RTS-CTS-DADOS-ACK) leva 3 segundos, ou seja, o computador que enviou a mensagem de reconhecimento da rede leva cerca de 6 segundos para receber uma resposta completa (reconhecimento e resposta do reconhecimento), continuando a ocasionar o *time-out*.

5.8.2 Comunicação utilizando o padrão Internet

Foi feito o teste para conferir o funcionamento da rede sem fio utilizando a rede Microsoft. Entretanto não estava sendo esperado o seu funcionamento porque o "Ambiente de Rede Local" do Windows não foi projetado para funcionar com uma baixa velocidade de transmissão.

Por este motivo no segundo teste o "Clientes para rede Microsoft " do Windows foi desinstalado, desligando o processo de *browsing* realizado para a identificação e comunicação entre os computadores ligados na rede Microsoft [7] [25]. Para permitir a comunicação utilizando o padrão Internet foi instalado no computador de teste com o Windows 98 o programa *Personal Web Server*.

A proposta para o funcionamento da rede no capítulo 4 está baseada no desenvolvimento de aplicações utilizando ambientes de desenvolvimento para Internet. Por este motivo este segundo teste foi dividido em duas partes: acesso a um página HTML simples e acesso a uma página HTML com *applet Java* trocando dados via *sockets*. O *applet* testado implementa um programa de *chat* simples.

O resultado obtido foi o tempo de 3 segundos para o envio completo de cada pacote (RTS-CTS-DADOS-ACK).

Não fez parte deste teste criar situações de forma a atenuar o sinal de rádio e forçar a ocorrência de erros de transmissão. Também não foi feito um estudo sobre o tamanho dos pacotes na transmissão. Foi escolhido como base o mesmo tamanho usado pelo protocolo AX25, ou seja pacotes com no máximo 256 bytes, também não foi feito um estudo sobre a performance relacionada ao tamanho do pacote. Como última consideração, o experimento

foi feito apenas utilizando dois computadores na transmissão, portanto o funcionamento da implementação para redes com três ou mais computadores não foi validado. Um estudo sobre a performance dos protocolos que fazem parte do IEEE 802.11 pode ser encontrado em [15].

5.9 Terceira versão do NDIS Miniport NIC driver

O DFWMAC possui um mecanismo opcional que envolve a troca de pacotes de controle (RTS/CTS) antes da transmissão do pacote de dados. Quando uma estação ganha acesso ao meio, ao invés de enviar imediatamente o pacote de dados, ela transmite um pacote de controle RTS. O RTS tem dois objetivos, reservar o meio para a transmissão dos dados e verificar se a estação destino está pronta para receber estes dados. Outra utilidade do RTS/CTS é para as situações onde uma estação não percebe o RTS enviado pelo transmissor, mas recebe o CTS enviado pelo receptor. Desta forma o canal, que não foi reservado pelo RTS, passa a ser reservado pelo CTS [2]. No experimento anterior optou-se pelo uso do RTS/CTS também para "economizar" o tempo das retransmissões ocasionadas por colisões. Como o pacote RTS é pequeno e rápido de ser transmitido, as colisões aconteceriam no envio do RTS, evitando a necessidade de retransmissão de pacotes de dados porque as colisões aconteceriam somente no pacote de controle RTS.

Nesta terceira versão do driver foi experimentado a retirada dos pacotes opcionais RTS/CTS. Implementando a transmissão apenas com o pacote de dados e o de controle ACK.

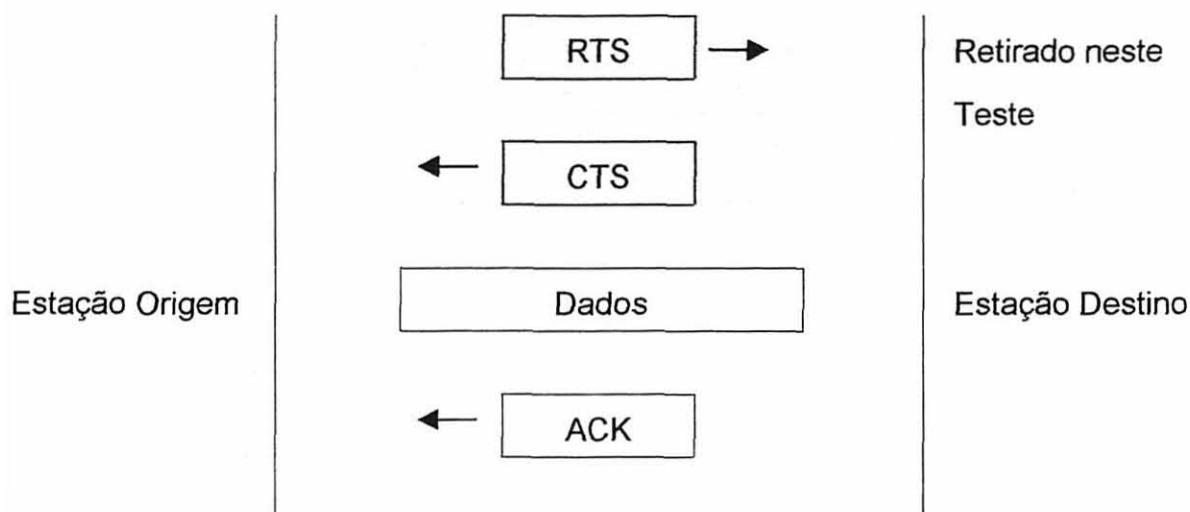


Figura 5.9 — Excluído os pacotes de controle opcionais nesta versão do driver

Esta retirada ocasionou a economia de 1,5 segundo para cada pacote transmitido. O resultado obtido foi o tempo de 1,5 segundo para o envio completo de cada pacote (DADOS-ACK).

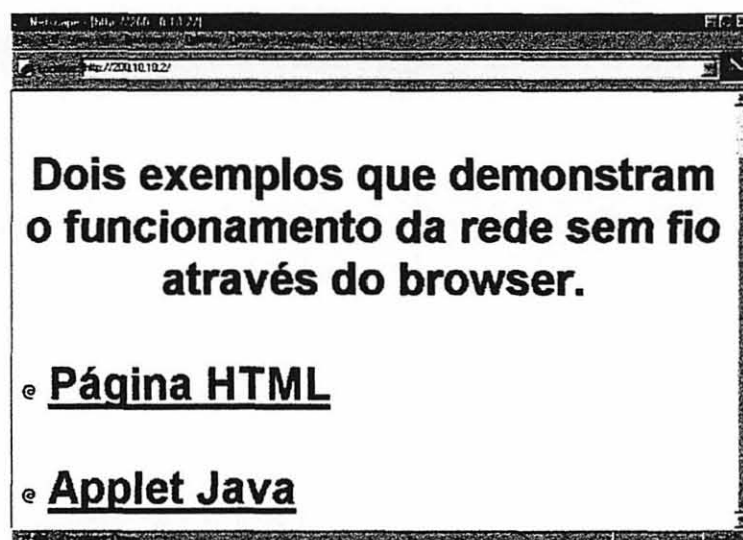


Figura 5.10 — Teste do funcionamento do driver com aplicações Internet

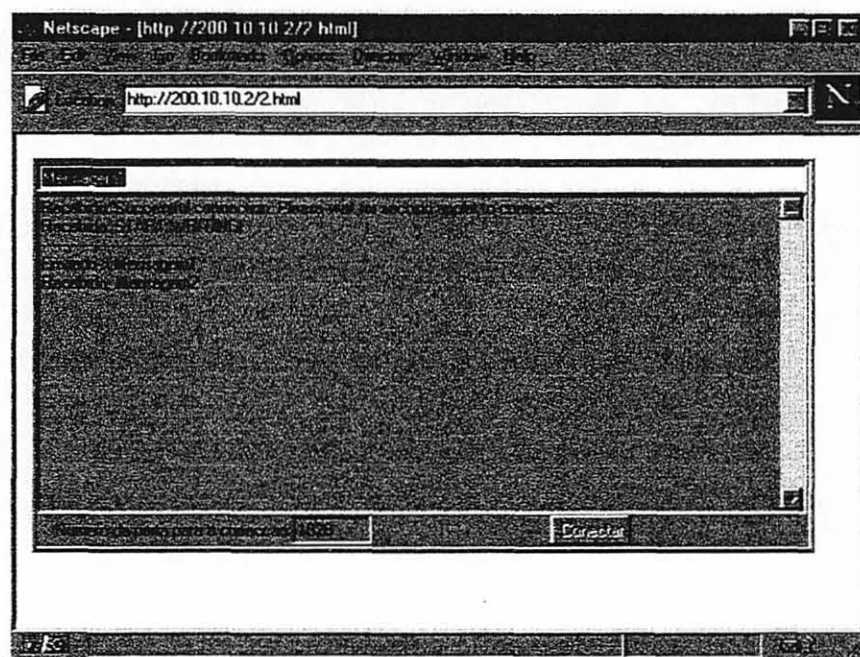


Figura 5.11 — Applet Java com um programa de chat simples

6. CONCLUSÃO E TRABALHOS FUTUROS

Para a área de comunicação de dados a melhor alternativa seria a existência de um único sistema que funcionasse sem fio, onde todas as aplicações pudessem ser implementadas neste sistema. Todavia, aplicações diferentes necessitam de soluções diferenciadas para que sejam economicamente viáveis. Este trabalho propõe uma rede sem fio que viabiliza o desenvolvimento de aplicações de monitoração remota que, em sua grande maioria não necessita transferir grandes quantidades de informação. Por exemplo, alguns sistemas de telesupervisão implantados na Rede Globo de São Paulo necessitam de apenas 200 bytes para informar a situação completa dos equipamentos existentes em uma estação monitorada. O total de estações monitoradas pelo computador central não passa de 10 estações.

Utilizando a plataforma de hardware/software proposta neste trabalho é possível desenvolver aplicações práticas que se beneficiam desta tecnologia como :

- *monitoração e controle de estações de trabalho remotas e de difícil acesso, situação muito comum em estações de rádio e TV que precisam monitorar/controlar transmissores que ficam em locais afastados;*
- *sistema que permite a polícia rodoviária ter acesso a banco de dados sobre pessoas procuradas, carros roubados, entre outros;*
- *segurança de empresas usando monitoração remota.*

Além da possibilidade de utilização desta tecnologia para o desenvolvimento de aplicações, este estudo sobre redes sem fio pode ser aprofundado para considerar outros temas, como por exemplo:

- utilização de mais de um servidor com o mesmo conceito de células da telefonia celular (topologia ESS proposta no IEEE 802.11);
- roteamento de IP para *hosts* móveis;
- utilização de diferentes protocolos de comunicação para obter uma melhor performance, como AX25 ou RPWNT [16];
- otimização do protocolo CSMA/CA, como o uso das mensagens do tipo *broadcast* e *multicast*;
- compactação de dados;
- segurança na transmissão dos dados através do uso de técnicas de criptografia;
- utilização de mais de uma frequência de rádio na comunicação;
- implementação utilizando hardware compatível com o padrão IEEE 802.11.

Apêndice A - DESCRIÇÃO DO HARDWARE UTILIZADO NO PROJETO

O trabalho para a programação das funções de manipulação do hardware utilizado no projeto (placa Bay 96) foi muito maior do que a programação da lógica de funcionamento do driver. A documentação sobre como programar/utilizar a placa é bastante reduzida, isso se deve ao uso bastante específico deste hardware em sua maioria por rádio-amadores. A documentação formal mais detalhada se refere ao CI Z8030/Z8530 contendo mais de cem páginas descrevendo o modo de funcionamento deste componente. Esta documentação foi fundamental durante o desenvolvimento do protótipo, porém o objetivo deste texto é descrever os circuitos e os modos/características de funcionamento. Apenas uma parte insuficiente desta documentação se referia a forma de programação do CI. Além disso, a documentação apresentava todos os recursos existentes no CI, mas estava fora do escopo explicar para que servia e quando cada uma destas características deveria ser usada. Outro detalhe, a documentação descrevia o modo de funcionamento de um CI, entretanto a placa utilizada no projeto possui dois deles. Como diferenciar cada um deles durante a programação?

Para entender a forma de programação deste hardware e até mesmo montar/soldar os cabos de conexão placa-rádio, foi necessário muita pesquisa, troca de mensagens pela Internet e a utilização do código fonte JNOS, um programa que implementa comunicação AX25 utilizando diversos tipos de placas seriais, uma delas com um projeto muito próximo da placa utilizada no protótipo.

Por isso este apêndice foi inserido nesta dissertação, colocando algumas das informações obtidas sobre a programação desta placa serial/rádio. Com certeza estas

informações não estão completas, mas é o mínimo necessário para a compreensão da documentação do Z8530, do código fonte JNOS e do código fonte escrito para o driver.

A1 Placa Bay 96 USCC

Neste trabalho foi utilizado como NIC uma placa serial tipo BayCom (BAY96_V2) da Microlaser Sistemas Optoeletrônicos. Esta placa é basicamente composta por quatro sessões [8][22]:

- **Sessão digital:** contém 2 CI's (Z8530) [24] controlados pelo barramento do PC. Estes CI's são dedicados à comunicação de dados no modo síncrono. Cada chip contém 2 ports provendo neste caso 4 canais de rádio.
- **Modem 3105:** o canal A do Z8530 é conectado diretamente ao modem TCM-3105. Este modem pode operar em VHF ou UHF conforme normal Bell202 em 1200 bauds. Alternativamente este chip poderá funcionar até 2400 bauds.
- **Modem 7911:** conectado ao canal B do Z8530, este modem poderá trabalhar em 1200 bauds (VHF ou UHF) ou em 300 bauds (HF), simplesmente mudando *straps* no corpo da placa.
- **Modem FSK:** este modem é compatível com o padrão G3RUH, pode funcionar entre 2400 a 38400 bauds, configurados por *straps* no corpo da placa.
- **Port livre:** conectado ao canal A do segundo Z8530, permite, através do *header disconnect*, interligar-se a outros modem's externos e compatíveis, podendo operar em múltiplas velocidades.

No protótipo foi utilizado o 4º canal da placa (USCC 03) que possui modem FSK podendo funcionar entre 2400 a 38400 bps. É utilizado apenas uma única frequência de rádio para a transmissão e para a recepção configurando uma transmissão do tipo *half-duplex*.

A2 Z8530

Como já foi apresentado a placa utilizada no projeto contém dois CI's Z8530 [24] (chamado de SCC - *Serial Communications Controller*) que podem ser programados diretamente pelo barramento do PC. Cada SCC funciona como um conversor/controlador serial-para-paralelo ou paralelo-para-serial. Cada SCC pode ser configurado por software para satisfazer uma grande variedade de aplicações seriais. O SCC possui diversos recursos como, por exemplo, mais de um tipo de protocolo, *baud rate generator* e gerador automático de CRC. Uma longa lista de registradores permite a manipulação dos seus recursos.

A2.1 Registradores

Cada canal do SCC possui 14 registradores de escrita, 7 registradores de leitura e mais 4 registradores que são compartilhados pelos dois canais do SCC, 2 para leitura e 2 para escrita. Dos registradores de escrita 10 são utilizados para controle, 2 para a geração de bytes de sincronização e 2 para a geração do *baud rate*. Os 2 registradores de escrita compartilhados são para a programação da interrupção e para o *reset* do SCC. Dos registradores de leitura, 4 deles contém informações sobre o estado do SCC, 2 contém dados sobre a geração do *baud rate* e um os dados recebidos. Os 2 registradores de leitura

compartilhados contém informações de controle sobre as interrupções pendentes e o vetor de interrupções.

É através destes registradores que o usuário consegue ler/enviar dados, programar os diversos modos de operação do SCC e fazer a leitura das informações de controle. Não foi colocado neste texto a lista dos registradores e as suas respectivas funções pois a documentação do Z8530 [24] apresenta nos capítulos 2, 7 e 8 uma extensa e detalhada apresentação sobre estes registradores.

Neste texto está sendo utilizado a seguinte notação para identificar os registradores: WRn para registradores de escrita e RRn para os registradores de leitura, onde "n" indicará o número do registrador.

A2.2 Modos de Operação

O Z8530 [24] disponibiliza dois canais *full-duplex* independentes. Estes canais podem ser utilizados para qualquer protocolo de comunicação síncrono ou assíncrono. O chip disponibiliza comunicação byte a byte síncrona e assíncrona, *monosync*, *IBM Bisync* e protocolos bit a bit como SDLC e HDLC.

A2.2.1 Modo Assíncrono

Existem vários modos de configuração para a comunicação assíncrona. Por exemplo: número de bits por símbolo, número de bits de parada, fator do *clock*, sinais da interface do modem e geração e detecção de *break*.

A2.2.2 Modo Síncrono

No modo de operação síncrono um padrão especial de bits é utilizado para disponibilizar a sincronização dos bytes. O chip possui várias opções de suporte ao modo síncrono incluindo várias quantidades de bits para os bytes de sincronização, número de bits por símbolo, geração e verificação de paridade, geração e verificação de CRC, assim como controles do modem.

A2.2.3 Modo SDLC

O modo SDLC [2] é útil para protocolos bit a bit. A placa não implementa o protocolo SDLC, a implementação SDLC da placa consiste apenas na técnica da inserção do "0" (*bit stuffing*) para alcançar a transparência na transmissão. No modo SDLC da placa os pacotes contendo informação são iniciados e terminados utilizando um único padrão de bits chamado de *flag*. O padrão 01111110 é utilizado como delimitador (*flag*). Para evitar a ocorrência dos delimitadores durante a transmissão, sempre que o transmissor encontrar cinco bits consecutivos iguais a 1 nos dados que vai transmitir, ele insere um bit 0 na cadeia de bits. Quando o receptor encontra cinco bits consecutivos, seguidos de um bit 0, ele automaticamente retira o bit 0.

Também está disponível no modo SDLC a geração automática de CRC ao final dos pacotes.

A3 Pinagem

Este projeto utilizou o USCC 03 da placa que possui um modem de 9600. Esta saída é identificada na placa através de um conector DB-25. A seguir é apresentado as conexões com as suas respectivas ligações ao rádio:

USCC		Rádio	
Pino	Descrição	Pino	Descrição
22	PTT	03	PTT
23	TX	04	Data in
24	RX	05	Data out
25	GND	2	GND
		1	12V DC

Figura A.1 — Pinagem ligando a placa USCC ao rádio

A4 Uso com o Rádio

De acordo com o fabricante da placa o tempo máximo que ela mantém a portadora (PTT) ativa é de 2000 milisegundos. Recomenda-se que o tempo de TX não seja superior a 500 milisegundos. O tempo para estabilização da portadora é de 30 milisegundos. Este rádio consegue manter com segurança o PTT ativo de acordo com a sua alimentação. Se estiver sendo alimentado com 9V, duas horas é o tempo máximo que pode manter a sua portadora. Para uma alimentação de 12V o tempo máximo de portadora é dois minutos. Por isso se estiver utilizando para a alimentação do rádio uma das saídas 12V do computador, recomenda-se o uso de um regulador para 9V, por exemplo um 7810.

A5 Endereçamento

No mercado existem várias placas diferentes que utilizam o Z8530 ou alguma de suas variações mais atuais. Cada uma destas placas possui algumas características diferentes em sua programação utilizando ou não os recursos oferecidos pelo CI. Uma das variações mais comuns está no número de CIs existentes na placa e na forma de endereçamento para acesso aos registradores.

A seguir é apresentado a forma de endereçamento da placa utilizada em nosso projeto, baseado no projeto da placa BayCom USCC [22].

A partir do JP4 da placa [8][22] é possível definir o seu endereço base:

Jumpers			Endereço
1	3	5	
Aberto	Aberto	Aberto	300h
Fechado	Aberto	Aberto	320h
Aberto	Fechado	Aberto	330h
Fechado	Fechado	Aberto	340h
Fechado	Aberto	Fechado	230h
Aberto	Fechado	Fechado	220h
Fechado	Fechado	Fechado	210h

Figura A.2 — Jumper para a seleção do endereço base da placa

Considerando os dois SCCs existentes na placa, configurados com o endereço base padrão (300h) resultaria na seguinte tabela de endereçamento:

Canal Placa	CI	Canal CI	End. Dados	End. Controle
USCC0	chip 01	A	300h	304h
USCC1	chip 01	B	301h	305h
USCC2	chip 02	A	302h	306h
USCC3	chip 02	B	303h	307h

Figura A.3 — Endereço de dados e de controle da placa USCC usando como endereço base o valor 300h

Como é possível notar na tabela apresentada o endereçamento muda de acordo com o número do canal USCC desejado. Existe uma conta que pode ser utilizada para calcular estes endereços:

$$\begin{aligned}\text{End. Dados} &= \text{EBase} + \text{NCI} * 2 + \text{NC} \\ \text{End. Controle} &= \text{End. Dados} + \text{OFF}\end{aligned}$$

Onde:

EBase → Endereço base da placa configurado nos *jumper*s.

NCI → Número do CI que deseja acessar (chip 01-0 ou chip 02-1).

NC → Número do canal (A-0 ou B-1).

OFF → Deslocamento para o registrador de controle do canal.

Exemplo: USCC3

$$\begin{aligned}\text{End. Dados} &= 300 + 1 * 2 + 1 = 303 \\ \text{End. Controle} &= 303 + 4\end{aligned}$$

Exemplo de um programa que calcula e guarda em uma tabela cada um destes endereços:

```
//Guarda o endereco da placa
Placa.nchips = 2;           //Num de chips
Placa.nchan  = 2;           //Num de canais por chip
Placa.iobase = 0x340;       //End base da placa
Placa.irq    = 7;           //Interrupcao da placa
Placa.clk    = 4915200L;    //Frequencia do PCLK/RTxC em Hz
Placa.clkmode= 16;          //Specifies the division rate (1,16,32)
Placa.pclk   = 1;
Placa.speed  = 9600;

//Calcula o enderecos dos regs de ctrl e dados de todos os canais
for (i=0; i < Placa.nchips; i++)
    for (j=0; j < Placa.nchan; j++) {
        Placa.chan[i][j].data = Placa.iobase + (i*Placa.nchan) + j;
        Placa.chan[i][j].ctrl = Placa.iobase + (i*Placa.nchan) + j + 4;
    }
```


A6 Acesso aos Registradores

Baseando-se no endereço base da placa e em qual canal se deseja trabalhar, a operação de manipulação dos registradores consiste em ler e escrever dados no endereço de controle (veja item **Endereçamento**). Por exemplo, considere o seguinte exemplo:

- endereço base da placa configurada para 300h;
- programação utilizando o quarto canal da placa (USCC3).

Neste exemplo o acesso as operações de *read* e *write* no registrador de controle deve acontecer no endereço 307h.

Como existe apenas um endereço para acesso aos registradores e mais de 20 registradores de leitura e escrita, a operação de manipulação dos registradores consiste em dois passos:

- seleção do registrador desejado;
- escrita/leitura do registrador.

A exceção a esta regra é o registrador de número 0, que pode ser acessado diretamente sem que seja necessário a seleção do registrador. A seguir é apresentado um exemplo de acesso a um registrador de escrita utilizando as funções *inportb* e *outportb* disponíveis na linguagem C:

```
outportb(CTRL,REG);      //Seleciona registrador
outportb(CTRL,VAL);      //Escreve valor no registrador
//CTRL -> corresponde ao endereço do registrador de controle
//      do canal desejado
//REG   -> corresponde ao número do registrador desejado
//VAL   -> corresponde ao valor a ser escrito no registrador REG
```

Outro exemplo com uma operação de leitura:

```

outportb(CTRL, REG);      //Seleciona registrador
VAL = inportb(CTRL);      //Leitura do registrador para vvel VAL

```

Mais um detalhe sobre as operações de leitura e gravação. Existe um tempo até que o registrador selecionado esteja disponível para acesso. Não respeitar este tempo fará com que a placa não funcione corretamente. Este tempo pode ter maior ou menor impacto na programação dependendo da velocidade do computador ou da forma como a placa está sendo programada, como por exemplo através uma sub-rotina em *assembler*. Por isso recomenda-se a inclusão de uma pausa entre a operação de seleção de registrador e da operação de leitura/escrita neste registrador. Exemplos:

```

//Escrita
outportb(CTRL, REG);      //Seleciona registrador
delay(1);                 //Pausa para aguardar a seleção do reg
outportb(CTRL, VAL);      //Escreve valor no registrador

//Leitura
outportb(CTRL, REG);      //Seleciona registrador
delay(1);                 //Pausa para aguardar a seleção do reg
VAL = inportb(CTRL);      //Leitura do registrador para vvel VAL

```

Como apresentado no início deste item, a exceção fica para o primeiro (número 0) registrador de gravação/leitura, onde a operação pode ser realizada diretamente:

```

outportb(CTRL, VAL);      //Escreve valor no registrador 0 (WR0)
VAL = inportb(CTRL);      //Leitura do registrador 0 (RR0) para vvel VAL

```

Agora um exemplo real de escrita nos registradores 0 e 3 (WR0 e WR3), lembrando que a operação será realizada no USCC3:

```

//Escrita registrador 0 (operação error reset)
outportb(0x307, 0x30);    //Escreve diretamente o valor no reg

//Escrita registrador 3 (operação disable rx)
outportb(0x307, 3);       //Seleciona o registrador 3
delay(1);                 //Pausa para aguardar a seleção do reg
outportb(0x307, 0);       //Escreve valor no registrador

```

Operação de leitura:

```

//Leitura registrador 0 (status register)
status = inportb(0x370);  //Operação usando apenas um passo

```

```
//Leitura registrador 3 (interrupt pending bits)
outportb(0x370, 3);      //Seleciona o registrador 3
delay(1);                //Pausa para aguardar a seleção do reg
IntBits = inportb(0x370); //Leitura do valor atual do registrador
```

Nem todos os registradores de escrita possuem os seus equivalentes registradores de leitura, ou seja, nem todas as configurações feitas através dos registradores de escrita podem ter os seus valores consultados. Por isso quando os registradores de escrita estão sendo usados é normal manter uma cópia do último valor escrito em cada registrador, permitindo assim que uma consulta, ou modificação possa ser feita posteriormente.

Foi usado um vetor para memorizar os últimos valores colocados nos registradores de escrita:

```
unsigned char wreg[20]; // Cópia dos valores escritos em cada reg
```

Assim, cada operação de gravação é guardada neste vetor:

```
//Escrita registrador 0 (operação error reset)
outportb(0x307, 0x30); //Escreve diretamente o valor no reg
wreg[0] = 0x30;         //Guarda valor para consulta futura

//Escrita registrador 3 (operação disable rx)
outportb(0x307, 3);     //Seleciona o registrador 3
delay(1);               //Pausa para aguardar a seleção do reg
outportb(0x307, 0);     //Escreve valor no registrador
wreg[27] = 0;           //Guarda valor para consulta futura
```

Este vetor com as cópias dos valores se torna realmente útil quando é necessário modificar apenas um dos bits de um determinado registrador:

```
//Escrita registrador 15 (external status interrupt control)
outportb(0x307, 15);   //Seleciona o registrador 3
delay(1);              //Pausa para aguardar a seleção do reg
wreg[49] &= DCDIE;     //Desliga apenas o bit que controla a
outportb(0x307, wreg[49]); //geração das ints de DCD
```

A partir da observação do código fonte do driver será possível verificar que algumas macros (*defines*) foram criadas para simplificar as operações de escrita/leitura,

automatizando as operações de seleção de registradores, *delay* e memorização dos valores no vetor.

A7 Programação da USCC

Como foi apresentado no item anterior (**Acesso aos Registradores**) é através da leitura e gravação nos registradores que se realiza a programação da USCC. Para a utilização da placa na comunicação de dados existem duas alternativas de programação: *polling* ou interrupção.

A7.1 Polling

Este é o modo mais simples de implementar. O programa deve constantemente consultar o SCC para saber se existem dados disponíveis, se novos dados podem ser enviados ou se alguma situação especial aconteceu, como um erro na recepção dos dados.

Para utilizar o modo de programação por *polling* o MIE (WR9 bit 3) e o Wait/DMA Request Enable (WR1 bit 7) devem estar configurados com o valor 0 de forma a desabilitar a ocorrência de interrupções pelo SCC. Com as interrupções desabilitadas cabe ao programa consultar o RR0 para identificar o estado do *buffer* de recepção, *buffer* de transmissão e do estado geral do SCC.

A7.2 Interrupção

Neste modo de programação o SCC irá gerar uma interrupção somente quando ocorrer alguma situação ou mudança significativa. Esta forma de programação permite que a CPU fique livre para outras atividades enquanto o SCC não está sendo utilizado. O SCC

tem um método interno de resolução de prioridades de forma a fazer com que a interrupção de maior prioridade no SCC seja tratada primeiro.

A8 Inicialização da Placa

A placa Bay 96 USCC contém 2 SCCs (Z8530) controlados pelo barramento do PC. Estes CI's são dedicados à comunicação de dados. Cada chip contém 2 *ports* provendo neste caso 4 canais de rádio. O modo mais simples de programação da placa é o modo assíncrono utilizando *polling*, o modo escolhido para a implementação do driver de rede foi o modo assíncrono, porém no lugar do *polling* será utilizado a programação por interrupções de forma a não sobrecarregar a CPU com a execução do *polling* pelo driver.

Para fazer o controle de cada um dos SCCs/canais existentes na placa foram definidas as seguintes estruturas de dados: USCCBay para controle da placa e a USCCChan para o controle de cada canal do SCC.

[illegible]

```

struct USCCChan {
    uint16 ctrl;          /* Endereco do reg de controle para este canal */
    uint16 data;          /* Endereco do reg de dados para este canal */
    unsigned char wreg[20]; /* Copia dos valores escritos em cada reg */
    long speed;           /* Velocidade em bps */
    char extclock;         /* External clock source on RTxC/TRxC */
    char fulldup;          /* External divider for fulldup available */
    char tx_inhibit;       /* Transmit is not allowed when set */
    /* statistic information on this channel */
    long ints;             /* Number of interrupts */
    long rxints;           /* Receiver interrupts */
    long txints;           /* Transmitter interrupts */
    long exints;           /* External/status interrupts */
    long spints;           /* Special receiver interrupts */
    long rxerrs;           /* CRC Errors */
    unsigned int rovers;   /* Receiver Overruns */
    unsigned int CRCFraming;
    unsigned char status; /* Status of the channel */
};

```

O processo de inicialização é dividido em três etapas. A primeira etapa consiste na programação do modo de operação (bits por símbolo, paridade, etc.) e carga das constantes (interrupção, *timer*). A segunda parte habilita as funções do hardware — transmissão, recepção, velocidade, entre outras. É importante que o modo de operação seja programado antes que as funções do hardware sejam habilitadas. A terceira parte, caso seja necessário, consiste em habilitar as diferentes interrupções.

Segundo o manual do Z8530 [24] a programação dos registradores no processo de inicialização deve ser feita da forma apresentada na figura A.4. Os bits nos registradores marcados com "X" devem ser programados pelo usuário. Os bits marcados com um "S" indicam que devem ser programados com os mesmos valores utilizados em passos anteriores.

Registrador	Bits	Comentários
Parte 1 — Modos e Constantes		
WR9	11000000	Hardware reset
WR4	XXXXXXXX	TX/RX com, Async or Sync mode
WR1	0XX00X00	Select W/REQ (optional)
WR2	XXXXXXXX	Program interrupt vector (optional)
WR3	XXXXXXXX0	Select RX control
WR5	XXXX0XXX	Select TX control
WR6	XXXXXXXX	Program sync character (optional)
WR7	XXXXXXXX	Program sync character (optional)
WR9	000X0XXX	Select interrupt control
WR10	XXXXXXXX	Miscellaneous control (optional)
WR11	XXXXXXXX	Clock control
WR12	XXXXXXXX	Time constant lower byte (optional)
WR13	XXXXXXXX	Time constant upper byte (optional)
WR14	XXXXXXXX0	Miscellaneous control
WR14	XXXSSSSS	Commands (optional)
Parte 2 — Habilitar		
WR14	000SSSS1	Baud Rate Enable
WR3	SSSSSSS1	RX enable
WR5	SSSS1SSS	TX enable
WR0	10000000	Reset TX CRG (optional)
WR1	XSS00S00	DMA Enable (optional)
Parte 3 — Interrupção		
WR15	XXXXXXXX	Enable external/status
WR0	00010000	Reset external status
WR0	00010000	Reset external status twice
WR1	SSSXXSXX	Enable RX, TX and Ext/Status
WR9	000SXSSS	Enable Master Interrupt Enable

Figura A.4 — Programação dos registradores para a inicialização da placa

Para exemplificar a inicialização é apresentado um trecho de código que faz a inicialização do canal 3 da placa para funcionar de forma assíncrona a 9600 bps. Este é um exemplo simplificado, não apresentando a configuração da rotina que fará o tratamento das interrupções já que isso é dependente dos recursos do sistema operacional. No código fonte do driver pode ser encontrado a rotina completa com a configuração das interrupções tanto para DOS como para Windows.

```

if (Placa.init != 0)
    return; //A placa ja foi inicializada

//Dados da placa
Placa.nchips = 2;           //Num de chips
Placa.nchan = 2;           //Num de canais por chip
Placa.iobase = 0x340;      //End base da placa
Placa.irq = 7;             //Interrupcao da placa
Placa.clk = 4915200L;      //Frequencia do PCLK/RTxC em Hz
Placa.clkmode= 16;         //Specifies the division rate (1,16,32)
                             //inside the SCC

Placa.pclk = 1;
Placa.speed = 49600;

//Calcula o enderecos dos regs de ctrl e dados de todos os canais
for (i=0; i < Placa.nchips; i++)
    for (j=0; j < Placa.nchan; j++) {
        Placa.chan[i][j].data = Placa.iobase + (i*Placa.nchan) + j;
        Placa.chan[i][j].ctrl = Placa.iobase + (i*Placa.nchan) + j + 4;
    }

//Resetar e pre-inicializar os chips da placa
RD(h,Placa.chan[0][A].ctrl,dummy); //Chip 1/2 channel A/B make
RD(h,Placa.chan[41][A].ctrl,dummy); //sure pointer is written
WCTRLRSCC(h,Placa.chan[0][A], R9,FHWRES); //Hardware reset chip 1/2
WCTRLRSCC(h,Placa.chan[41][A], R9,FHWRES); //channel A/B

for (i=0; i < 1000; i++) dummy = i+1; //wait a while to be sure

//Inicializa cada um dos canais
for (i=0; i < Placa.nchips; i++)
    for (j=0; j < Placa.nchan; j++) {
        RD(h,Placa.chan[i][j].ctrl,dummy); //make sure pointer is written
        WCTRLRSCC(h,Placa.chan[i][j],R4,0); //no mode selected yet
        WCTRLRSCC(h,Placa.chan[i][j],R1,0); //no W/REQ operation
        WCTRLRSCC(h,Placa.chan[i][j],R2,
            ((unsigned char)(16 * i))); //chip# in upper 4 bits of
                                         //vector
        WCTRLRSCC(h,Placa.chan[i][j],R3,0); //disable rx */
        WCTRLRSCC(h,Placa.chan[i][j],R5,0); //disable tx */
        WCTRLRSCC(h,Placa.chan[i][j],R9,VIS); //vector includes status, MIE
                                         //off
    }

Placa.init = TRUE; //Placa foi inicializada com sucesso;

```



```

WCTRLRSCC(h, Placa.chan[41][B], R4, X16CLK|SB1); /*16 clock, 1 stopbit
WCTRLRSCC(h, Placa.chan[41][B], R1, 0);          // no W/REQ operation
WCTRLRSCC(h, Placa.chan[41][B], R3, Rx8|RxCRC_ENAB); //RX 8 bits/char, CRC,
                                                //disabled
WCTRLRSCC(h, Placa.chan[41][B], R5, Tx8|DTR|TxCRC_ENAB); //TX 8 bits/char, CRC,
                                                //DTR, disabled
WCTRLRSCC(h, Placa.chan[41][B], R9, VIS);          //vector includes status */
WCTRLRSCC(h, Placa.chan[41][B], R10, NRZ|0);       //select NRZ
WCTRLRSCC(h, Placa.chan[41][B], R14, 0);
WCTRLRSCC(h, Placa.chan[41][B], R11, RCTRxCp|TCRTxCp); /* */
WCTRLRSCC(h, Placa.chan[41][B], R14, DISDLL);      /* */

//Set SCC channel speed
//clkmode specifies the division rate (1,16,32) inside the SCC
{ uint brgrate;
  long spdclkm;

  //Calculate baudrate generator value
  spdclkm = Placa.speed * Placa.clkmode;

  brgrate = (unsigned)
    ((Placa.clk + spdclkm) /
     (spdclkm * 2)) - 2;
  CCTLRSCC(h, Placa.chan[41][B], R14, BRENABL); //disable baudrate
generator
  WCTRLRSCC(h, Placa.chan[41][B], R12,
    (unsigned char)brgrate);          //brg rate LOW
  WCTRLRSCC(h, Placa.chan[41][B], R13,
    (unsigned char) (brgrate >> 8)); //brg rate HIGH
  OCTLRSCC(h, Placa.chan[41][B], R14, BRENABL); //enable baudrate generator
}
WCTRLRSCC(h, Placa.chan[41][B], R15, BRKIE|TxUIE|DCDIE); //enable ABORT, DCD
and
                                                //HUNT/SYNC interrupt
WR(h, Placa.chan[41][B].ctrl, RES_EXT_INT);      //reset ext/status interrupts
WR(h, Placa.chan[41][B].ctrl, RES_EXT_INT);      //must be done twice
RD(h, Placa.chan[41][B].ctrl, dummy);            //read initial status
//enable interrupts
OCTLRSCC(h, Placa.chan[41][B], R1, INT_ALL_Rx|TxINT_ENAB|EXT_INT_ENAB);
OCTLRSCC(h, Placa.chan[41][B], R9, MIE);         //master interrupt enable

```

A9 Controle das interrupções

Cada canal do SCC contém 3 fontes de possíveis interrupções, como cada SCC possui dois canais são 6 as possíveis fontes de interrupção geradas pelo SCC. As 3 possíveis fontes de interrupção de cada canal são as seguintes:

- recepção;
- transmissão;
- External ou Status.

A9.1 Recepção

As interrupções de recepção podem ser geradas porque um byte está disponível ou porque uma condição especial aconteceu. A interrupção relativa a recepção de dados acontece quando o byte já foi carregado para o FIFO e está pronto para ser lido. As condições especiais podem ser FIFO *overrun*, CRC/erro no pacote, final de pacote e erro de paridade. O erro de paridade pode ser incluído como condição especial dependendo do valor configurado no bit 2 do WR1. O estado das condições especiais podem ser lidos através do RR1.

O modo como as interrupções de recepção serão geradas é controlado através da configuração dos bits 3 e 4 do WR1. É possível selecionar um dos três modos disponíveis:

- interrupção desabilitada;
- interrupção no primeiro byte ou condição especial;
- interrupção em todos os bytes e condições especiais.

A9.2 Transmissão

A interrupção de transmissão tem apenas uma fonte. É gerada somente quando o *buffer* de transmissão passa de cheio para vazio. Esta interrupção será gerada somente após o envio do primeiro byte. É através do uso desta interrupção que os pacotes de dados são enviados.

As interrupções de transmissão são controladas a partir do *Interrupt Enable bit*, bit 1 do WR1.

Outra forma de controlar as transmissões sem o uso da interrupção pode ser feita através do RR0, que contém um bit indicando se o *buffer* de transmissão está ou não vazio.

A9.3 External/Status

Existem diversas fontes de interrupção *external/status* que podem ser geradas. Esta geração é configurada através do WR15. O controle deste tipo de interrupção pode ser ligado/desligado através do bit 0 do WR1.

As configurações possíveis são as seguintes:

- zero count;
- DCD;
- Sync/Hunt;
- Transmitter underrun;
- Break/Abort.

A9.4 Identificando o motivo da interrupção

A placa possui 2 SCCs Z8530 compartilhando a mesma interrupção que pode ser configurada por *jumper* existente na placa [8][22]. Se na inicialização da placa foi escolhido o método de programação através de interrupção (veja item **Programação da USCC**), a cada geração de interrupção é necessário identificar a qual SCC/canal esta interrupção pertence, e qual é o seu tipo. Esta interrupção pode indicar uma recepção, transmissão ou *external/status*. O processo é feito em dois passos a partir da verificação dos registradores RR2 e RR3.

O código apresentado a seguir é executado pela rotina chamada a cada interrupção gerada pela placa USCC.

```

i = 0;
while ((i < 2) && (watchdog-->0)) {                                     (1)
    RR(h,Placa.chan[i][A].ctrl,R3,dummy);                               (2)
    if (dummy != 0) {
        canal = (dummy < 8) ? B : A;
        RR(h,Placa.chan[i][B].ctrl,R2, IsrMotivoIRQ);                   (3)
        Ne2000IsrMotivoIRQ &= 0x6; //Isola informação de status do vetor
        Ne2000IsrMotivoIRQ >>= 0x1; //Descarta o bit menos significativo
                                   //conforme tab 7-4 (7-11) do manual 8530

        □
        //00 -> TX (Transmit Buffer Empty)                               |Chan B
        //01 -> EX (External Status Change)                             |
        //10 -> RX (Receive Character Available)                         |
        //11 -> SX (Special Receive Condition)                           |
        switch (Ne2000IsrMotivoIRQ)
        {
            case 2:
                Placa.chan[i][canal].rxints++;
                //Rotina para tratamento do RX
                break;
            case 0:
                Placa.chan[i][canal].txints++;
                //Rotina para tratamento do TX
                break;
            case 1:
                Placa.chan[i][canal].exints++;
                //Rotina de tratamento do EX
                break;
            case 3:
                Placa.chan[i][canal].spints++;
                //Rotina de tratamento do SX
                break;
        } //switch
        i = 0;                                                            (4)
    } //if
    else {
        i++;                                                              (5)
    }
} //while

```

(1) while ((i < 2) && (watchdog-->0)) {

Neste *while* é controlado a verificação do motivo da interrupção, o valor 2 indica que a verificação será feita nos dois SCCs disponíveis na placa. A variável *watchdog* é usada para controlar um possível erro que possa acontecer com a placa, como por exemplo,

parar de responder. Neste caso, é evitado que a execução do programa trave dentro da interrupção.

```
(2) RR(h, Placa.chan[i][A].ctrl, R3, dummy)
```

Independente do canal que gerou a interrupção este comando deve ser executado no canal A do SCC. O valor retornado contém nos bits a indicação do canal que gerou a interrupção (conforme página 7-23 figura 7-21 no manual Z8530).

```
(3) RR(h, Placa.chan[i][B].ctrl, R2, IsrMotivoIRQ)
```

O último passo consiste em verificar o motivo da interrupção (transmissão, *external status*, recepção ou condição especial). A tabela indicativa dos bits se encontra na página 7-11 tabela 7-4 do manual do 8530.

```
(4) i = 0
```

No caso de alguma interrupção tenha sido indicada pelo RR3, deve-se voltar a verificação desde o início (1º SCC) já que é possível que durante o tratamento da interrupção, alguma outra situação de interrupção tenha sido sinalizada.

```
(5) i++
```

Caso nenhuma interrupção tenha sido sinalizada para este SCC, a variável é incrementada de forma a fazer a verificação do próximo SCC disponível na placa.

A10 Envio de dados

Os dados podem ser enviados utilizando duas formas, ou através do endereço de dados (veja item **Endereçamento**) ou através do registrador WR8.

Exemplo para o USCC3 com endereço base 0x300, utilizando o registrador:

```
//Enviando dados pela USCC
outportb(0x307,8);          //Seleciona o registrador 8
delay(1);                  //Pausa para aguardar a seleção do reg
outportb(0x307,'A');        //Envia byte
```

Usando diretamente o endereço de dados:

```
//Enviando dados pela USCC
outportb(0x303,'A');        //Envia byte
```

Através desta operação é possível enviar um byte. Para enviar mais de um byte, durante a inicialização da placa deve-se escolher como será o modo de programação: *polling* ou interrupção.

A10.1 Polling

O envio de um grupo de bytes por *polling* funciona da seguinte forma:

- 1 - Enviar o primeiro byte do grupo
- 2 - Enquanto falta bytes a enviar
 - 3 - Aguardar o *buffer* de transmissão ficar vazio
 - 4 - Envia próximo byte

- 1 - Enviar o primeiro byte do grupo

Este primeiro passo pode ser realizado de uma das duas formas apresentadas:

```
//Enviando dados pela USCC
outportb(0x303,'A');        //Primeiro byte
```

- 3 - Aguardar o *buffer* de transmissão ficar vazio

Esta verificação pode ser feita utilizando o registrador RR0, através do bit 2:

```
//Busy waiting Tx Buffer Empty
while (!(inportb(0x370)& 0x4));
```

- 4 - Envia próximo byte

Envia o próximo byte do grupo repetindo a operação apresentada no passo 1.

A10.2 Interrupção

A interrupção de transmissão é gerada somente quando o *buffer* de transmissão passa de cheio para vazio. Esta interrupção é gerada somente após o envio do primeiro byte. É através do uso desta interrupção que os pacotes de dados são enviados. Primeiramente coloca-se o primeiro byte do pacotes no *buffer* de transmissão do SCC, logo que o SCC gerar uma interrupção de Transmissão significa que o próximo byte a ser transmitido pode ser colocado no *buffer* de transmissão e assim sucessivamente até o último byte do pacote. Neste momento a placa deve ser sinalizada de que não existem mais bytes a serem enviados. Exemplo:

Envio do primeiro byte dando início ao processo de interrupções:

```
outportb(0x303, 'A');      //Primeiro byte
```

Conforme mostrado no item **Identificando o motivo da interrupção** quando uma interrupção de TX for gerada, basta enviar o próximo byte. Quando não existirem mais bytes a serem enviados, é preciso executar a seguinte linha de comando (operação com o WR0):

```
outportb(0x307, 0x28);    // Reset TxINT Pending
```

A partir deste comando é encerrado o processo de geração de interrupção de envio, finalizando o processo de envio deste pacote de dados.

A11 Recepção de dados

Assim como no envio, a recepção de dados também pode fazer uso das duas formas: endereço de dados e registrador RR8.

Exemplo para o USCC3 com endereço base 0x300, utilizando o registrador:

```
//Faz a leitura do byte da USCC
outportb(0x307,8);      //Seleciona o registrador 8
delay(1);               //Pausa para aguardar a seleção do reg
c = inportb(0x307);      //Byte recebido
```

Usando diretamente o endereço de dados:

```
//Faz a leitura do byte da USCC
inportb (0x303);        //Byte recebido
```

A maneira que a USCC avisa da existência ou não de um byte também está associado ao modo de programação escolhido (*polling* ou interrupção).

A11.1 Polling

É possível verificar se um byte está disponível através do registrador RR0. O primeiro bit (bit 0) indicará se existe ou não um byte disponível:

```
if (inportb(0x307) & 0x1) //Existe byte?
    C = inportb (0x303);    //Byte recebido
```

O SCC possui um FIFO para a recepção dos bytes com um tamanho de 3 bytes. Portanto a operação de verificação e leitura de bytes precisa ser rápida de forma a evitar *overruns*.

A11.2 Interrupção

Se durante o processo de inicialização da placa foi escolhido o modo interrupção, a cada byte recebido a USCC irá gerar uma nova interrupção. Dentro deste interrupção é feito a leitura do byte que foi recebido (veja item **Identificando o motivo da interrupção**).

A operação de leitura é a mesma utilizada no *polling* com exceção da verificação da existência de byte no *buffer*:

```
C = inportb (0x303);      //Byte recebido
```


Foi mencionado que não é necessário verificar a existência de bytes no *buffer*, todavia como o FIFO de recepção de bytes possui 3 bytes de profundidade, dependendo da velocidade que a rotina de interrupção leva para identificar e tratar a interrupção, pode ser possível que outro byte já tenha sido recebido pela USCC. Uma alternativa é a utilização de uma repetição durante a leitura dos bytes dentro da interrupção:

```
do {  
    *cp++ = inportb(0x303);    //Proximo byte do FIFO  
} while (inportb(0x307) & 0x1); //Mais bytes?
```

A12 Modo Loopback

Assim como os modems o SCC disponibiliza o modo *loopback* para testes. O uso deste modo permite que a programação da comunicação seja feita sem a utilização de dois computadores/placas, simulando o envio e a recepção de dados usando o mesmo computador.

Para ativar o modo *loopback* durante a inicialização do canal o bit 4 do WR14 deve ser ativado.

Apêndice B - ETHERNET

B1 Ethernet

Ethernet é uma das mais populares tecnologias utilizadas atualmente. Outros tipos de LAN incluem *Token Ring*, *Fast Ethernet*, *Fiber Distributed Data Interface* (FDDI), *Asynchronous Transfer Mode* (ATM) e *LocalTalk*. *Ethernet* é uma camada física de rede popular porque possui um bom balanço entre velocidade de transmissão, custo e facilidade de instalação.

Hoje o padrão *Ethernet* é definido pelo *Institute for Electrical and Electronic Engineers* (IEEE) como IEEE 802.3. Este padrão define as regras para a configuração e como os elementos em uma rede do tipo *Ethernet* assim como os elementos da rede interagem entre si. [44] [47]

O início do desenvolvimento do *Ethernet* foi feito pela Xerox que a registrou como uma marca sua. A tecnologia foi refinada e uma segunda geração chamada de *Ethernet II* foi definida, sendo bem aceita pelo mercado. Esta atualização da *Ethernet* também é conhecida como DIX e teve a participação da Digital, Intel, and Xerox.

Para que este padrão pudesse se tornar internacional o IEEE recebeu a tarefa de desenvolver um padrão para todas as tecnologias de LANs. Com este objetivo foi montado o comitê "802" para estudar *Ethernet*, *Token Ring* e outras tecnologias LAN. O objetivo do projeto não foi apenas padronizar cada um dos tipos de LAN individualmente, mas também estabelecer regras que poderiam ser globais para todos os tipos de LANs. Desta forma os dados de uma rede *Ethernet* poderiam mais facilmente passar para redes *Token Ring*.

Esta visão ampla criou conflitos entre a prática *Ethernet* existente e o antigo sistema Xerox DIX. O IEEE foi cuidadoso em separar as novas e velhas regras. Reconheceu que haveria um período onde mensagens DIX e as novas mensagens IEEE 802 teriam que coexistir na mesma LAN.

B2 Camadas Ethernet

Duas camadas, física e de enlace, estão envolvidas na especificação *Ethernet*. A camada física realiza as seguintes funções, que são geralmente associadas ao controle físico do canal:

- **Codificação dos dados**, o que inclui a geração e a remoção do preâmbulo para sincronização e a codificação e decodificação dos bits.
- **Acesso ao canal**, que inclui a transmissão e a recepção dos bits codificados, verificação da portadora e detecção de colisões.

As duas principais funções da camada de enlace são as seguintes:

- **Encapsulamento dos dados**, que inclui a montagem dos pacotes, manipulação dos endereços de origem e destino e detecção de erros no canal físico.
- **Gerenciamento do link**, que inclui a alocação do canal, o processo de evitar colisões e o tratamento das colisões ocorridas.

B3 Ethernet X IEEE 802.3

Antes do desenvolvimento de um padrão internacional, a Xerox administrava as convenções do *Ethernet*. Cada empresa que adotou o *Ethernet* desenvolveu um protocolo diferente, por este motivo dois bytes foram adicionados neste padrão de forma a identificar

o tipo do protocolo. Dentro do processo de identificação de colisão no protocolo *Ethernet* um pacote pequeno tem bytes acrescentados de forma a atingir o mínimo de 64 bytes [45]. Esta regra fez com que os protocolos de alto nível precisassem possuir ou um tamanho mínimo obrigatório nas mensagens, ou um campo interno que permitisse diferenciar os bytes de dados dos bytes adicionados para completar o tamanho mínimo da mensagem.

Alguns tipos definidos pela Xerox:

- 0x0600 XNS (Xerox)
- 0x0800 IP (the Internet protocol)
- 0x6003 DECNET

O comitê IEEE 802 recebeu a tarefa de desenvolver protocolos que poderiam trabalhar da mesma forma entre todas as LANs. Como já foi mencionado, para evitar colisões, uma rede Ethernet de 10 megabits precisa ter pacotes com o tamanho mínimo de 64 bytes. Qualquer mensagem menor do que isso precisa ser completada com zeros. A necessidade de controlar uma mensagem com tamanho mínimo é único para a Ethernet e não se aplica a outros meios utilizados nas LANs. Para fazer com que a Ethernet pudesse trocar mensagens com outros tipos de LANs, seria necessário dispor de um campo "tamanho" para permitir a distinção entre os bytes de dados e os bytes utilizados para completar a mensagem.

O padrão DIX (*Ethernet II*) não precisava de um campo tamanho porque os protocolos desenvolvidos pelos fabricantes (XNS, DECNET, IPX, IP) todos tinham em seu próprio protocolo um campo identificando o tamanho da mensagem. Todavia, o comitê 802 precisava de um padrão que não dependesse do comportamento dos protocolos de alto

nível. Por isso o padrão 802.3 trocou os dois bytes do campo "tipo", que fazem parte do *Ethernet II*, por um campo, também de dois bytes, mas que representam o tamanho da mensagem.

Esta modificação foi possível porque nenhum tipo importante definido pela Xerox utilizou valores menores do que 1500. Como o tamanho máximo de um pacote *Ethernet* contém até 1500 bytes, não haveria conflito entre o DIX e o padrão 802. Desta forma, qualquer pacote *Ethernet* com um campo tipo/tamanho menor ou igual a 1500 é do formato 802.3 (o campo identifica tamanho) enquanto qualquer pacote contendo um valor maior do que 1500 identifica o formato DIX (campo identifica o tipo).

Por isso é possível encontrar os dois tipos de pacotes trafegando por uma rede *Ethernet* sem que exista conflito. O TCP/IP utiliza o padrão DIX original da Xerox, enquanto o NetBEUI utiliza o padrão 802.3.

B4 Formato do pacote

Os pacotes *Ethernet* possuem os seguintes campos apresentados na figura B.1[45]:

Pacote	Descrição
62 bits	Preâmbulo Uma série de 62 bits alternando 1's e 0's utilizado para assegurar a sincronização com o receptor.
2 bits	Delimitador do início do pacote Dois bits 1 consecutivos. O preâmbulo e o delimitador de início juntos geram os seguintes bits: 10101010 10101010 10101010 101010101 10101010 10101010 10101010 10101011
6 bytes	Endereço de destino Este campo de 48 bits especifica para qual estação o pacote está endereçado. O primeiro bit indica o tipo de endereço. Se for 0 o endereço corresponde a uma

	única estação na rede, se for 1 indica que o endereço especifica um grupo lógico de estações. Por exemplo todos os bits com o valor 1 sinalizam uma mensagem do tipo <i>broadcast</i> .
6 bytes	Endereço de origem 48 bits indicando um único endereço que corresponde a estação que enviou a mensagem.
2 bytes	Tamanho ou Tipo Para o IEEE 802.3 estes 16 bits mostram a quantidade de bytes que estão sendo transmitidos (campo dados). Para <i>Ethernet I</i> ou <i>II</i> (DIX) este campo indica o tipo do pacote. Os códigos de tipo são maiores que 1500 para permitir identificar se ele é um tipo ou um tamanho. Por exemplo pacotes IP são do tipo 0x800.
46 a 1500 bytes	Dados O padrão IEEE 802 recomenda que o campo dados tenha um tamanho de 46 a 1500 bytes. Se o pacote de dados é menor do que 46 bytes, para que ele funcione corretamente no protocolo <i>Ethernet</i> , uma quantidade de bytes serão adicionados ao final do campo de dados para que atinja o tamanho mínimo de 46 bytes. Isso permite diferenciar pacotes válidos de colisões.
4 bytes	CRC Esta parte do pacote contém 32 bits calculados utilizando o polinômio AUTODIN II (normalmente gerado pelo chip). O CRC inclui o endereço de destino, endereço de origem e o campo com os dados.

Figura B.1 — Campos do pacote Ethernet

O menor pacote é composto de 60 bytes ($6 + 6 + 2 + 46 = 60$ bytes) e o maior pacote 1514 bytes ($6 + 6 + 2 + 1500 = 1514$ bytes).

B5 Endereço Ethernet

Os 48 bits utilizados como endereço pelo *Ethernet* é chamado de *Organizationally Unique Identifier* (OUI) e tem o objetivo de criar um endereço único entre todas as placas de rede. O bit 47 indica o tipo de endereço. Se for 0 o endereço corresponde a uma única estação na rede, se for 1 indica que o endereço especifica um grupo lógico de estações. Por exemplo todos os bits com o valor 1 sinalizam uma mensagem do tipo *broadcast*. O bit 46

Apêndice C - FERRAMENTAS UTILIZADAS NO PROJETO

Para o trabalho no protótipo foram necessários a utilização softwares, bibliotecas e códigos fonte. Estão enumerados neste apêndice os que foram mais importantes.

C1 Microsoft SDK/DDK

Como o protótipo foi desenvolvido para a plataforma Windows o uso do SDK/DDK disponibilizado pela Microsoft foi fundamental. Este pacote compreende a documentação referente ao desenvolvimento de drivers. Conforme mostra a documentação da Microsoft, o driver foi desenvolvido de acordo com as regras do Windows NT de forma a permitir a sua instalação no Windows 95/98/NT. No protótipo o funcionamento do driver foi testado apenas com o Windows 95 e com o Windows 98.

De forma complementar a documentação SDK/DDK existe uma série de fontes exemplificando como os diversos tipos de drivers devem ser programados. Para a programação do driver foi utilizado como base o programa exemplo da placa NE2000. Este pacote contém ainda as bibliotecas/ferramentas necessárias para permitir a compilação/link do driver.

C2 Softlce NuMega

A programação de um driver não é uma operação trivial. A sua ligação direta com o sistema operacional acrescenta um número grande de detalhes que precisam ser atendidos durante a programação. Por esse motivo durante o aprendizado da programação de um driver muitos detalhes são entendidos apenas quando observado o comportamento do driver durante a sua execução. A utilização de um *debugger* pode auxiliar neste processo,

permitindo que o programa seja executado passo a passo e os valores/fluxo de execução possam ser observados. Os ambientes de desenvolvimento já incorporam poderosas ferramentas de *debug*, todavia como programação de um driver não é uma tarefa comum aos programadores, estas ferramentas de *debug* não dão suporte a estes tipos de programa. Estas operações de *debug* ficam ainda mais difíceis quando a programação é de um *kernel driver*, como um driver de rede. Neste tipo de programação não é permitida operações de interface, como a leitura de dados do teclado ou a apresentação de janelas.

Para permitir a possibilidade de *debug* de um driver existe a ferramenta chamada *SofIce*, onde a operação de *debug* é feita dentro de um ambiente especial, utilizando-se dois computadores ligados pela serial. Um computador possui o driver que deve estar compilado com o modo *debug* ativado. O outro computador é usado para visualizar as linhas do código que estão sendo executadas. A utilização de uma ferramenta do tipo *SofIce* durante a construção de um driver é quase obrigatória, já que a ferramenta de *debug* que acompanha o SDK não funciona facilmente em drivers do tipo *kernel*, precisando de versões de *debug* específicas do sistema operacional.

C3 Borland C++ 3.1

A programação para Windows ou a programação de um driver acrescenta um número muito grande de detalhes. Quando o objetivo é entender o funcionamento de um código, ou experimentar a programação de uma placa, os detalhes do sistema operacional podem dificultar ainda mais esta tarefa. Por este motivo foi escolhido aprender sobre o funcionamento da placa a partir do desenvolvimento de programas DOS. Assim é possível

concentrar nos detalhes de funcionamento da placa e não nas particularidades do sistema operacional. Para construção destes programas de teste foi utilizado o Borland C++.

C4 Visual C++ 6.0

Para a programação do driver não foi utilizado o ambiente de desenvolvimento Visual C++ apenas o seu compilador de linha. O SDK/DDK contém somente os *scripts* necessários para a compilação, por isso é necessário a existência de um compilador instalado.

C5 Word 97

Usado para a montagem deste texto.

C6 NDIScope for Windows

Este programa pode ser encontrado no site www.pcausa.com (*Printing Communications Assoc., Inc.*) e permite a monitoração dos *pacotes* que estão trafegando pela rede. Este programa foi utilizado para verificar se o pacote enviado por uma estação estava sendo recebido corretamente pela estação destino. Alguns erros na montagem do pacote pela estação receptora foram detectados a partir da comparação dos pacotes apresentados neste programa.

C7 BayCom

Como aprender a utilizar um hardware sem ter a certeza de que ele está funcionando corretamente? Este trabalho iniciou do zero, com a montagem e soldagem dos cabos de conexão ao rádio. Durante o desenvolvimento aconteceram alguns problemas relativos ao funcionamento do hardware. Depois de diversas semanas de trabalhos perdidos a estratégia

de teste do driver foi modificada. Sempre que alguma modificação na configuração do hardware ou a substituição de algum componente da placa era feito o programa Baycom (conhecido pelos rádio-amadores) era usado para verificar se o hardware estava funcionando corretamente. Somente depois desta verificação é que os teste com o protótipo eram reiniciados.

C8 JNOS

Como aprender o funcionamento de um hardware sem a existência de um programa exemplo? Felizmente o programa JNOS é distribuído com o código fonte. Este programa não é específico para o hardware utilizado no projeto, mas pode ser configurado para ser usado com placas USCC padrão Baycom. A partir do estudo deste código fonte, que entre módulos, arquivos de cabeçalho e configuração possui mais de 300 arquivos, foi possível entender o princípio de funcionamento da USCC.

C9 Sourcer

Esta é uma antiga ferramenta que a partir de um programa executável um programa assembler equivalente é gerado. Esta ferramenta foi utilizada algumas vezes para verificar o funcionamento de algumas partes do programa Baycom. O Baycom não possui código fonte aberto.

C10 Adobe Acrobat Reader

Alguns documentos encontrados na Internet estavam no formato PDF.

C11 GSVIEW32 2.1 - A Ghostscript graphical interface

Utilizado para a leitura de alguns artigos encontrados na Internet.

C12 Terminal do Windows

O processo de *debug* de um driver é diferente do processo tradicional de *debug* de um programa escrito para Win32. Para complementar e muitas vezes simplificar o funcionamento do *SoftIce*, já que a versão do *SoftIce* utilizada no projeto era de avaliação (30 dias), foi implementado algumas operações de *debug* utilizando diretamente a porta serial do computador. Assim, algumas mensagens de *debug* do driver (aquelas como "passei por aqui", ou "o valor atual é...") foram "impressas" usando a porta serial e um computador adicional com um programa terminal para visualização destas mensagens.

Apêndice D - SCRIPT PARA INSTALAÇÃO DO DRIVER DE REDE

Não existe na documentação da Microsoft referência de como construir um *script* para a instalação de um driver de rede. A documentação apresenta apenas as diretrizes gerais de como construir *scripts* para a instalação de dispositivos. O *script* apresentado a seguir foi feito com base no *script* para a instalação da NE2000. Esta referência foi retirada do CD de instalação do Windows 95, mas também é compatível com o Windows 98.

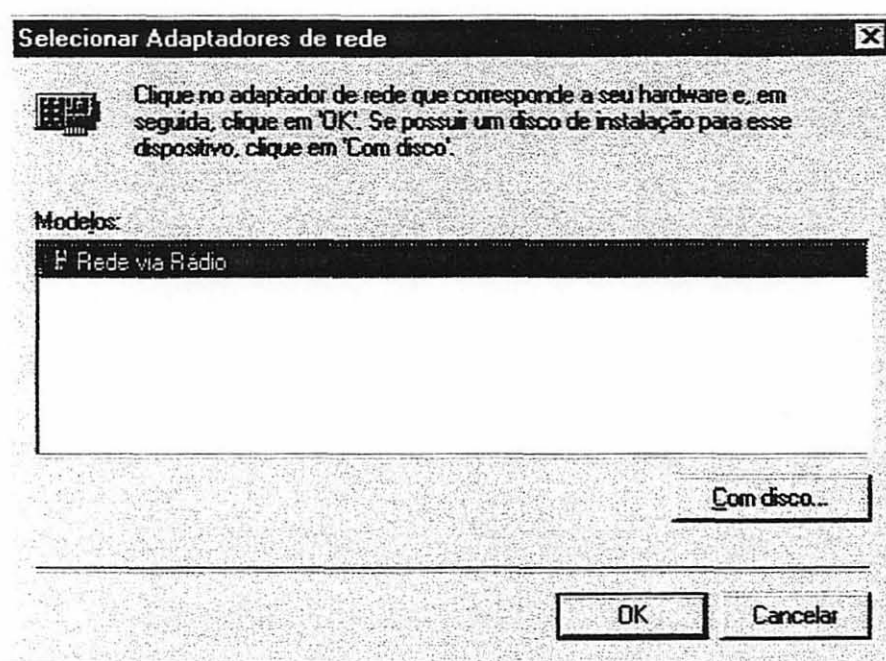


Figura D.1 - Janela do Windows 98 para adicionar o driver da rede sem fio

```
[Version]
Signature=$CHICAGO$
Class=Net
Provider=%V_PROVIDER%

[manufacturer]
%V_MANUFACT% = parau_drivers

[parau_drivers]
%*RADIO1.desc% = RADIO1_ndi, *RADIO1

[RADIO1_ndi]
AddReg=RADIO1_ndi_reg
DeviceID=RADIO1
```

```

MaxInstance=1

[RADIO1_ndi_reg]
HKR,Ndi,DeviceID,,RADIO1

;->from Netnovel.inf
HKR,,DevLoader,,*ndis
HKR,,DeviceVxDs,,radio.sys
;->end from Netnovel.inf

;Parâmetros específicos do driver
HKR,, Address,,340
HKR,, IRQ,,7

; NDIS Info
HKR,NDIS,LogDriverName,,RADIO1
HKR,NDIS,MajorNdisVersion,1,03
HKR,NDIS,MinorNdisVersion,1,0A
HKR,NDIS\NDIS3,DriverName,, "RADIO1$"
HKR,NDIS\NDIS3,FileName,, "*RADIO1"
; Interfaces
HKR,Ndi\Interfaces,DefUpper,, "ndis3"
HKR,Ndi\Interfaces,DefLower,, "serial"
HKR,Ndi\Interfaces,UpperRange,, "ndis3"
HKR,Ndi\Interfaces,LowerRange,, "serial"
; Install sections
HKR,Ndi\Install,ndis3,, "RADIO.Install"
HKR,Ndi\Remove,ndis3,, "RADIO.Remove"

[RADIO.Install]
AddReg=RADIO.AddReg
CopyFiles=RADIO.CopyFiles

[RADIO.AddReg]
HKR,,DevLoader,,*ndis

[RADIO.CopyFiles]
RADIO.SYS

[RADIO.Remove]
AddReg=RADIO.Rmv.AddReg
CopyFiles=RADIO.Rmv.CopyFiles

[RADIO.Rmv.AddReg]

[RADIO.Rmv.CopyFiles]
RADIO.SYS

[DestinationDirs]
DefaultDestDir  =11 ; LDID_SYS
RADIO.CopyFiles  =11

[strings]
V_MANUFACT="Parau UFPR"
V_PROVIDER="Parau UFPR"
*RADIO1.desc="Rede via Rádio"

```

Apêndice E - REPRODUÇÃO DO EXPERIMENTO PRÁTICO

- 1) Preparar os cabos para a ligação da placa ao rádio (veja Apêndice A).
- 2) Configurar dois computadores com o sistema operacional Windows 95 ou 98.
- 3) Configurar a placa para a IRQ 7 e o ADDR 0x340 [8].
- 4) Instalar placas verificando conflitos de IRQ e endereçamento com outros dispositivos.
- 5) Verificar o funcionamento da placa/rádio/cabos usando o programa Baycom.
Para conectar utilize o comando `CONN <nome estação> 3`.
- 6) Instalar o driver no Windows (use *script* do Apêndice D). Uma forma alternativa para a instalação do driver da rede sem fio é o de instalar a placa NE2000 e depois substituir o arquivo NE2000.SYS pelo arquivo do driver da rede sem fio (Capítulo 5).
- 7) Desinstalar do Windows o "Clientes para rede Microsoft" de forma a desligar o processo de *browsing* (Capítulo 5).
- 8) Instalar e configurar o protocolo TCP-IP.
- 9) Reinicializar o computador.

Para que a visualização de páginas HTML funcione é necessário a instalação e configuração do *Personal Web Server*.

REFERÊNCIAS BIBLIOGRÁFICAS

Livro/Manual:

- [1] BAKER, Art. **The Windows NT Device Driver Book**. Prentice Hall. 1997.
- [2] COLCHER, Sérgio, LEMOS, Guido, SOARES, Luiz Fernando Gomes. **Redes de Computadores**. Das LANs, MANs e WANs às Redes ATM, Segunda Edição. 1995.
- [3] DHAWAN, Sanjay. **Networking Device Drivers**. VNR (Van Nostradan Reinhold) International Thomson Publishing Inc. 1995.
- [4] GEIER, Jim. **Wireless LANs: Implementing Interoperable Networks**. Macmillan Technical Publishing. 1999.
- [5] GIBILISCO, Stan. **Handbook of Radio and Wireless Technology**. McGraw-Hill. 1998.
- [6] LAVERGHETTA, Thomas S. **Microwaves and Wireless Simplified**. Artech House Publishers. 1998.
- [7] **Microsoft Developer Network Library**. October 1997.
- [8] **Manual de Hardware BAY96_V2**. Microlaser Sistemas Optoeletrônicos Microcomputação e Comunicação.
- [9] **Networking Basics: Self-Study Training Kit**. Microsoft Corporation. 1994.
- [10] STALLINGS, William. **Operating Systems Internals and Design Principles**. Chapter 8/11. Prentice Hall. 1998.

- [11] TANENBAUM, Andrew S. **Modern Operating Systems**. Chapter 10. Prentice Hall. 1992.
- [12] TISCHER, Michael. **PC Internt System Programming**. Chapter 7. Abacus. 1995.
- [13] VISCAROLA, Peter G., MASON, W. Anthony. **Windows NT Device Driver Development**. OSR Open System Resources, Inc. Macmillan Technical Publishing. 1999.

Artigo:

- [14] BALAKRISHNAN, Hari, PADMANABHAN, Venkata N., SESHAN, Srinivasan, KATZ, Randy H. **A Comparison of Mechanisms for Improving TCP Performance over Wireless Links**. Computer Science Division, Department of EECS, University of California at Berkeley.
- [15] CHHAYA, S. Harshal, GUPTA, Sanjay. **Performance modeling of asynchronous data transfer methods of IEEE 802.11 MAC protocol**. Wireless Networks Journal No 3. Baltzer Science Publishers. August 1997.
- [16] EWERTON, Peres Cesário. **Estudo do Desenvolvimento de Protocolos de Rede em Ambiente Windows NT - Microsoft**. Comunicação interna. UFPR. 2000.
- [17] FORMAN, Gerooge H., ZAHORJAN, John. **The Challenges of Mobile Computing**. Computer Science and Engineering. University of Washington. March 9, 1999.
- [18] TOURRILHES, Jean. **The improvement offered by PiggyData and Packet Frame**. Packet Frame Grouping works. Proc. of PIMRC '98.

- [19] TOURRILHES, Jean. **Robust Broadcast: Improving the Reliability of Broadcast Transmissions on CSMA/CA**. Personal Systems Laboratory. HP Laboratories Bristol. HPL-98-38. February, 1998.

Especificação:

- [20] **IEEE 802.11 : Wireless LAN medium access control (MAC) and physical layer (PHY) specifications**.IEEE.
- [21] IEEE P802.15 **Working Group for Wireless Personal Area Networks (WPANs)**.
Grouper.ieee.org/groups/802/15/pub/wpan-faq.htm
- [22] **The modular BayCom USCC-Card: USCC>4**.

<http://www.baycom.org/bayweb/news/uscc4.htm>
- [23] **The Official Bluetooth Website**. www.bluetooth.com
- [24] **ZiLOG Serial Communication Controllers Products**. ZiLOG Technical Manual
Z8030/Z8530 SCC Serial Communication Controllers.

www.zilog.com/products/scc.html

Revista:

- [25] **Configuring Browser Options for WinNT, Win95 and WFW**. Exploring Windows NT, June 1996. The Cobb Group. Cópia do artigo em

www.microsoft.com/TechNet/winnt/ntwrkstn/Tips/ewn9662.asp.
- [26] DEVONEY, Chris. **A Nação sem Fio**. Windows Computing – Pág. 52. Julho de 1994.

[27] HARVEY, David A., SANTALES, Richard. **A Comunicação sem Fio Torna-se Realidade**. Byte Junho 1994, Pág. 65 - 69.

[28] **Indústria Aposta nas redes Wireless**. LanTimes Brasil. Junho 1997 Vol. 3 Edição 4
Pág 28.

[29] SOUZA , Paulo Henrique de. **Infocomunicação Pessoal por Wireless**. Network World Dezembro 1998.

Revista na Internet:

[30] **AOL terá acesso à Web por celular**. Info Online - 7/02/00.

<http://www2.uol.com.br/info/infonews/022000/07022000-14.shl>

[31] **Brasil terá portal para transmissão wireless**. Terça-feira, 11 de janeiro de 2000 -
09h17. Infonews. <http://www2.uol.com.br/info/infonews/012000/11012000-0.shl>

[32] CRESPO, Rose. **Rede Tess vai transmitir dados**. 20/10/99. Info Online - Plantão
Info. www.infoexame.com.br

[33] CRESPO, Rose. **Celular do futuro**. - Linha Cruzada - Info Online - No 16 - 15/12/99.
www.infoexame.com.br

[34] ELIAS, Jô. **IDC prevê que 61 milhões de usuários usarão a Web sem fio em 2003**.
IDG Now - 9/02/00. www.uol.com.br/idgnow/inet/inet2000-02-08f.shl

[35] FUOCO, Taís. **Vésper realiza primeira chamada via WLL**. Plantão Info - Info
Online - 21/10/99. www2.uol.com.br/info/infonews/101999/21101999-11.shl

- [36] GARCIA, Karina. **Internet via celular chega ao Brasil.** - IDG Now - 10/02/00.
www.uol.com.br/idgnow/telco/telco2000-02-09e.shl
- [37] **Usuários de TDMA já são 30 milhões no mundo.** 01/12/99. Info Online - Plantão
 Info. www.infoexame.com.br
- [38] **Wireless: preço ainda não é competitivo.** ComputerWorld – Julho 1998.
http://www.uol.com.br/computerworld/computerworld/260/tele_05.htm
- [39] **Wireless.** IDG. http://idg.uol.com.br/idgnow/2000/especial_tecnologias
 Internet:
- [40] **1895: Marconi's Invention.** www.alpcom.it/hamradio/marconi.html
- [41] BRUNERO, Angelo, VALORI, Andrea. **The Invention of the Radio.**
www.alpcom.it/hamradio/radeng.html, (IK1QLD Angelo Brunero and IK1QFT
 Andrea Valori).
- [42] **Comunicação de Dados Via Celular.** Trellis Produtos para Comunicação de Dados
 Ltda. 1999. <http://www.trellis.com.br/celular.html>
http://www.uol.com.br/networkworld/networkworld/nwt05/techup_02.htm
- [43] COUEY, Anna. **The Birth Of Spread Spectrum.**
<http://www.sirius.be/lamarr.htm>
- [44] DANG, Hui. **Origin and History of Ethernet.**
<http://bugs.wpi.edu:8080/EE535/hwk96/hwk2cd96/dang/dang.html>

- [45] **Ethernet Specification.** WPI EE535 Telecommunications and Transmission Technologies Course Homepage.

<http://bugs.wpi.edu:8080/EE535/hwk3cd95/harveywl.html>

- [46] **Network – Air Interfaces.** Lucent Technologies.

www.lucent.com/wireless/products/networks/airinter.html

- [47] PCLT -- H. Gilbert. **Ethernet.** 12 Apr 1995.

<http://pclt.cis.yale.edu/pclt/COMM/ETHER.HTM>

- [48] **Selecting wireless LAN technology.** Proxim.

<http://www.proxim.com/learn/whiteppr/select.shtml>

- [49] SILVA, Adailton J. S. **Redes Wireless.** RNP. 1999.

<http://vexxor.virtualave.net/redes/wireless.htm>

- [50] TAPR archives. **AX.25 Amateur Packet-Radio Link-Layer Protocol Version 2.2.**

November 1997. <http://www.tapr.org/tapr/html/ax25.html>

- [51] **What is a wireless LAN?** <http://www.proxim.com/learn/>

- [52] **What is W-CDMA.** NEC.

http://www.nec.com.sg/nems/wcdma/whatis/what_bot.htm